

hyväksymispäivä arvosana

arvostelija

## **Sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittely**

Miia Nopanen

Helsinki 13.5.2016

HELSINGIN YLIOPISTO  
Tietojenkäsittelytieteen laitos

Tiedekunta – Fakultet – Faculty		Laitos – Institution – Department	
Matemaattis-luonnontieteellinen tiedekunta		Tietojenkäsittelytieteen laitos	
Tekijä – Författare – Author			
Miia Nopanen			
Työn nimi – Arbetets titel – Title			
Sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittely			
Oppiaine – Läroämne – Subject			
Tietojenkäsittelytiede			
Työn laji – Arbetets art – Level	Aika – Datum – Month and year	Sivumäärä – Sidoantal – Number of pages	
Pro gradu -tutkielma	13.5.2016	89 sivua + 9 liitesivua	
Tiivistelmä – Referat – Abstract			
<p>Tutkielma on luonteeltaan kirjallisuuskatsaus, ja se käsittelee sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittelyä. Sen yhtenä tavoitteena on selvittää vaatimusmäärittelyn rooli tietojärjestelmäkehityksessä. Työn aikana selvisi, että vaatimusmäärittely on erittäin keskeinen osa tietojärjestelmien kehittämistä. Sen tuloksia hyödynnetään kaikissa kehitystyön vaiheissa. Ne vaikuttavat erittäin paljon siihen, onnistutaanko kehitysprojektissa tuottamaan toimiva ja käyttäjien tarpeet täyttävä tietojärjestelmä.</p> <p>Toisena tavoitteena on selvittää sähköisen terveydenhuollon tietojärjestelmien erityispiirteet. Tutkielmassa tunnistettiin sähköisen terveydenhuollon tietojärjestelmien erityispiirteiksi järjestelmien turvallisuuskriittisyys, korostunut tarve tietosuojalle järjestelmissä käsiteltävien tietojen sensitiivisyyden vuoksi, tarve yhteentoimivuudelle muiden tietojärjestelmien kanssa, voimakas sidonnaisuus lakeihin ja asetuksiin sekä moninaiset sidosryhmät omine intresseineen ja tarpeineen.</p> <p>Edellä esitetystä heräsi kiinnostus, kuinka sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittely kannattaisi suorittaa. Selvisi, ettei ole olemassa mitään mallia, joka ohjeistaa huomioimaan edellä mainitut erityispiirteet ja tukee vaatimusmäärittelyn onnistumisesta. Tutkielman kontribuutio-osuudessa luodaan malli sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittelyn suorittamiseen. Malli koostuu kolmesta erilaisesta vaatimusmäärittelyn iteraatiovariaatiosta, joissa on huomioitu sähköisen terveydenhuollon erityispiirteet. Lisäksi niihin on sisällytetty erilaisia vaatimusmäärittelyn onnistumista mahdollistavia ja tukevia seikkoja, kuten vaatimusmäärittelytyön onnistumistekijöitä ja parhaita käytäntöjä. Mallia voidaan soveltaa erilaisissa tietojärjestelmäprojekteissa valitusta systeemityömenetelmästä riippumatta.</p> <p>ACM Computing Classification System (CCS) 2012:  Health care information systems  Health informatics  Life and medical sciences  Management of computing and information systems  Software creation and management</p>			
Avainsanat – Nyckelord – Keywords			
vaatimusmäärittelyt, systeemityö, sähköinen terveydenhuolto, telelääketiede, sähköiset lääkemääräykset			
Säilytyspaikka – Förvaringställe – Where deposited			
Kumpulan tiedekirjasto			
Muita tietoja – Övriga uppgifter – Additional information			

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Vaatimusmäärittely tietojärjestelmäkehityksessä</b>	<b>5</b>
2.1	Tietojärjestelmät ja niiden kehittäminen	5
2.2	Vaatimusmäärittelyprosessin työvaiheet	10
2.3	Vaatimusmäärittelyssä tunnistettujen vaatimusten luokitteluja	17
2.4	Vaatimusmäärittelyn laatu ja onnistuminen	23
<b>3</b>	<b>Sähköinen lääkemääräys sähköisen terveydenhuollon esimerkkitaapauksena</b>	<b>32</b>
3.1	Sähköisen terveydenhuollon historia ja erityispiirteet	32
3.2	Sähköinen lääkemääräys ja sen hyödyt ja ongelmat	38
3.3	Sähköisen lääkemääräyksen tietojärjestelmäratkaisut	41
3.4	Suomen sähköisen lääkemääräyksen toiminnallisuudet	46
<b>4</b>	<b>Malli sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittelyyn</b>	<b>49</b>
4.1	Vaatimusmäärittelyn esivalmistelut	50
4.2	Vaatimusmäärittelyn ensimmäinen iteraatiovariaatio	52
4.3	Vaatimusmäärittelyn toinen iteraatiovariaatio	61
4.4	Vaatimusmäärittelyn kolmas iteraatiovariaatio	70
4.5	Iteraatiovariaatioiden sijoittuminen erilaisissa kehitysprojekteissa	74
<b>5</b>	<b>Sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittely</b>	<b>76</b>
<b>6</b>	<b>Yhteenveto</b>	<b>81</b>
	<b>Lähteet</b>	<b>85</b>
	<b>Liite 1 Vaatimusmäärittelyn onnistumisen osatekijät</b>	
	<b>Liite 2 Suomen sähköisen lääkemääräyksen toiminnallisuudet</b>	
	<b>Liite 3 Karkeistettu vaatimusmäärittelymalli</b>	
	<b>Liite 4 Vaatimusmäärittelyn iteraatiovariaatiot vesiputousmallissa</b>	

# 1 Johdanto

Tietotekniikka ja sen avulla tuotetut ratkaisut ovat kiinteä osa nykyajan yhteiskuntaa. Tietoteknisiä ratkaisuja hyödynnetään jo lähes kaikilla aloilla, kuten maataloudessa, kaupankäynnissä ja opetuksessa. Hyvin voimakkaasti kasvava sovellusalue on terveydenhuolto [Thü15, 5-12]. Kun väestön keski-ikä nousee, on terveydenhuoltoalan pystyttävä palvelemaan entistä suurempaa joukkoa kansalaisia. Lisäksi nykytrendien mukaisesti terveydenhuollon palveluiden tulisi olla saatavilla mahdollisimman yksilöllisesti ja kustannustietoisesti, ajasta ja paikasta riippumatta. Jotta kasvaviin tarpeisiin voitaisiin vastata, on tarkoituksenmukaista valjastaa tietotekniikka ja automaatio helpottamaan, nopeuttamaan ja varmistamaan terveydenhuollon monimutkaisia toimintaprosesseja.

Terveydenhuoltoalan tietoteknisistä ratkaisuista käytetään usein termiä telelääketiede tai sähköinen terveydenhuolto (engl. digital health, e-Health tai telehealth). Tässä tutkielmassa käytetään pääsääntöisesti termiä sähköinen terveydenhuolto. Määritelmän alaisuuteen kuuluu joukko erilaisia diagnostisia ja interventionaalisia teknologioita, jotka perustuvat ICT-alan saavutusten kuten digitaalisten viestintäverkkojen, tietojenkäsittelyn ja tekoälyn hyödyntämiseen [Thü15, 13-16].

Konkreettisina esimerkkeinä sähköisen terveydenhuollon tietojärjestelmistä ja sovelluksista voidaan mainita muun muassa terveyteen liittyvät älypuhelimien mobiilisovellukset, kannettavat insuliinipumput ja päätöksentekoa tukevat potilastietojärjestelmät. Teknologioiden kehitystyö jatkuu edelleen, ja arvioiden mukaan vuonna 2017 maailmassa tullaan käyttämään noin 18 miljardia euroa terveydenhuoltoalan ICT-kustannuksiin. Tulevaisuudessa tullaan varmasti näkemään uusia terveydenhuoltoalaa edistäviä innovaatioita esimerkiksi biometriikkaan, kvanttilaskentaan ja 3D-tulostukseen liittyen.

Tämän pro gradu -tutkielman aiheena on sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittely. Työ on luonteeltaan kirjallisuuskatsaus eikä se sisällä varsinaista empiiristä osuutta. Työn lähteinä on käytetty oppikirjoja perusymmärryksen luomiseksi sekä alan tutkijoiden julkaisemia artikkeleita ja konferenssijulkaisuja ymmärryksen laajentamiseksi ja uusimpien tutkimustulosten selvittämiseksi.

Tutkielmaa ohjaavat tutkimuskysymykset ovat seuraavat:

1. Mitä erityispiirteitä sähköisen terveydenhuollon tietojärjestelmissä on?
2. Mikä rooli vaatimusmäärittelyllä on tietojärjestelmäkehityksessä?
3. Miten sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittely kannattaa suorittaa?

Ensimmäiseen tutkimuskysymykseen liittyen tutkielmassa tunnistettiin viisi erilaista sähköisen terveydenhuollon tietojärjestelmien erityispiirrettä. Ensimmäinen erityispiirre on järjestelmien turvallisuuskriittisyys (engl. safety critical systems). Turvallisuuskriittisyydellä tarkoitetaan sitä, että mahdolliset häiriöt tietojärjestelmässä voivat vahingoittaa ympäristöään ja käyttäjiänsä ja johtaa pahimmassa tapauksessa kuolemaan [Kni02, 547]. Tämä tulee esille esimerkiksi leikkaussalin robotiikassa tai sairaudenhoitovälineiden teknologioissa, kuten insuliinipumpun tai sydämentahdistimen mikroprosessoreissa. Tällaisten järjestelmien tulee toimia erityisen turvallisesti ja luotettavasti. Toinen tunnistettu erityispiirre oli se, että niissä käsitellään sensitiivisiä, yksittäisten henkilöiden terveydentilaa koskevia tietoja [PKK15, 49 ja BrH15, 134]. Erittäin keskeisessä roolissa onkin tietojärjestelmin tietoturva ja yksityisyyden suoja.

Kolmas erityispiirre oli yhteentoimivuus (engl. interoperability) muiden tietojärjestelmien kanssa. Modernit toimintatavat sekä tarve tarjota palveluita ajasta ja paikasta riippumatta tarkoittaa sitä, että potilaiden tietoja tulee voida välittää joustavasti eri laitteiden, järjestelmien ja organisaatioiden välillä [BLC15, 95-97]. Tällainen tiedonvälitys edellyttää järjestelmien yhteentoimivuutta, jossa olennaisena tekijänä ovat yhteiset standardit. Neljänneksi erityispiirteeksi tunnistettiin voimakas sidonaisuus lakeihin ja asetuksiin [PKK15, 48-49]. Sähköisen terveydenhuollon tietojärjestelmien kehitystyössä tulee aina selvittää ja huomioida paikallinen lainsäädäntö niin terveydenhuollon kuin tuoteturvallisuuden ja tietosuojankin osalta. Viides erityispiirre on se, että sähköisen terveydenhuollon järjestelmiin liittyy usein monia heterogeenisiä sidosryhmiä [VFS15, 157-159]. Sidosryhmät voivat olla järjestelmän käyttäjätahoja, rahoittajia tai poliittisia päättäjiä, joilla on joku rooli uuden järjestelmän käytössä tai hallinnoinnissa.

Jotta sähköisen terveydenhuollon tietojärjestelmät toimisivat toivotulla tavalla ja niissä huomioitaisiin edellä mainitut erityispiirteet, on järjestelmien kehitystyössä panostettava vaatimusmäärittelyyn. Toisen tutkimuskysymyksen osalta tultiinkin siihen päätelmään, että vaatimusmäärittely on systeemityön ja tietojärjestelmän elinkaaren ensimmäinen ja tietyllä tavalla tärkein vaihe [Lam09, 42-43]. Vaiheen aikana kootut toiminnalliset ja ei-toiminnalliset vaatimukset toimivat pohjana tietojärjestelmän suunnittelu- ja toteutustyölle. Lisäksi niitä vasten suoritetaan järjestelmän testaus, jolla todennetaan, toimiiko järjestelmä tilaajan toivomalla tavalla. Vaatimusmäärittelyyn tulisi panostaa kaikissa tietojärjestelmien kehitysprojekteissa. Tutkimusten mukaan epäonnistunut vaatimusmäärittely ja epäselvät vaatimukset ovat yksi yleisimmistä syistä koko tietojärjestelmäprojektien epäonnistumiseen [The95, 4-5 ja FGZ15, 26].

Tutkielmassa tutustuttiin sähköisen terveydenhuollon tapausesimerkkinä sähköiseen lääkemääräykseen. Työn edetessä todettiin, ettei Suomen sähköisen lääkemääräysjärjestelmän vaatimusmäärittelyn suorittamisesta ollut saatavilla tarkempia kuvauksia. Ilmeisesti koko toimialalla ei ole olemassa mallia tai sapluunaa, kuinka sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittely tulisi suorittaa. Tämän havainnon pohjalta nousi kolmas tutkimuskysymys, jonka puitteissa haluttiin selvittää, miten sähköisen terveydenhuollon tietojärjestelmän vaatimusmäärittely kannattaa suorittaa.

Selvitystyön tulokset koottiin vaatimusmäärittelytyön malliksi, joka pohjautuu van Lamsweerden esittelemään vaatimusmäärittelyprosessiin ja sen vaiheisiin [Lam09, 34]. Mallissa hyödynnetään van Lamsweerden määrittelemiä vaatimusmäärittelydokumentaation laatutavoitteita [Lam09, 35-36], El Emamin ja Madhavjin tunnistamia vaatimusmäärittelyn onnistumiseen vaikuttavia osatekijöitä [EIM95, 208-209] sekä Hofmanin ja Lehnerin kuvaamia vaatimusmäärittelyn parhaita käytäntöjä [HoL01, 65-66]. Kolmesta erilaisesta iteraatiovariaatiosta koostuvan mallin avulla suoritettua vaatimusmäärittelyä avulla sähköisen terveydenhuollon tietojärjestelmien kehitystyöllä on ennistä paremmat edellytykset onnistua ja tuottaa käyttäjien tarpeet täyttävä tietojärjestelmä.

Tutkielma on suunnattu ensisijaisesti henkilöille, jotka työskentelevät sähköisen terveydenhuollon tietojärjestelmien kehittämistehtävissä tai osallistuvat niiden vaatimusmäärittelyyn. Vaatimusmäärittelyn ammattilaiset saavat lisäymmärrystä siitä, millaisia eri-

tyispiirteitä nimenomaan sähköisen terveydenhuollon vaatimusmäärittelytyössä on huomioitava. Vaatimusmäärittelytyöhön osallistuvat liiketoiminnan edustajat puolestaan saavat käsityksen siitä, mistä vaatimusmäärittelyssä on ylipäänsä kyse ja miten he voisivat omalta osaltaan edesauttaa vaatimusmäärittelyn onnistumista. Saatuja tuloksia voidaan siis hyödyntää uusien sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittelytyön suunnittelussa ja suorittamisessa.

Tutkielma jakautuu kahteen osioon; teoriaosuuteen ja siihen pohjautuvaan kontribuutioosuuteen. Luvussa 2 perehdytään ylätasolla tietojärjestelmien kehittämiseen ja syvennytään tarkemmin vaatimusmäärittelyprosessiin. Samalla siinä selvitetään, mitkä tekijät vaikuttavat vaatimusmäärittelyn laatuun ja tukevat sen onnistumista. Luvussa 3 luodaan katsaus sähköisen terveydenhuollon historiaan ja nykytilanteeseen. Samalla selvitetään, millaisia erityispiirteitä näillä tietojärjestelmillä on ja esitellään sähköinen lääkemääräys yhtenä sähköisen terveydenhuollon tapausesimerkkinä. Luvussa 4 kuvataan tutkielman teoriaosuuteen pohjautuva vaatimusmäärittelymalli. Malli koostuu kolmesta erilaisesta iteraatiovariaatiosta, joiden avulla vaatimusmäärittelytyötä tehdään iteratiivisesti tarkentaen. Luvussa myös kerrotaan, kuinka mallia voidaan hyödyntää vesiputousmallia tai Scrumia noudattavissa kehitysprojekteissa. Luvussa 5 kootaan vastaukset tutkimuskysymyksiin ja pohditaan, miten niitä voitaisiin hyödyntää tulevaisuudessa sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittelyssä ja sen jatkokehityksessä. Lisäksi siinä tuodaan esille joitakin jatkotutkimusaiheita. Lukuun 6 on koottu lyhyt yhteenve-to tutkielman keskeisistä tuloksista.

## 2 Vaatimusmäärittely tietojärjestelmäkehityksessä

Tässä luvussa avataan yleisellä tasolla tietojärjestelmä-käsitettä ja perehdytään tietojärjestelmien kehittämiseen. Siinä luodaan myös teoriapohja vaatimusmäärittelylle esittelemällä vaatimusmäärittelyprosessi työvaiheineen sekä prosessin aikana tunnistettavien vaatimusten luokitteluita. Lopuksi kerrotaan vaatimusmäärittelyn laadusta ja tekijöistä, jotka vaikuttavat sen onnistumiseen. Ensisijaisina lähteinä käytetään Risto Pohjosen näkemyksiä tietojärjestelmistä ja niiden kehittämisestä [Poh05] sekä Axel van Lamsweerden perusteosta vaatimusmäärittelystä [Lam09]. Vaatimusmäärittelyn laatua ja onnistumista tarkastellaan erityisesti El Emamin ja Madhavjin esittelemien vaatimusmäärittelyn onnistumistekijöiden [ELM95] sekä Hofmannin ja Lehnerin esittelemien vaatimusmäärittelyn parhaiden käytäntöjen [HoL01] kautta.

### 2.1 Tietojärjestelmät ja niiden kehittäminen

Tietojärjestelmien (engl. information system) avulla pyritään mahdollistamaan, helpottamaan tai tehostamaan jotakin toimintaa. Termi tietojärjestelmä voidaan ymmärtää useilla eri tavoilla, ja tässä tutkielmassa tukeudutaan Risto Pohjosen [Poh05, 338] esittelemään määrittelyyn. Sen mukaan tietojärjestelmällä tarkoitetaan tietojenkäsittely- ja tiedonsiirtolaitteista, ohjelmista, tiedoista, ohjeista sekä näitä käyttävistä ihmisistä muodostuvaa kokonaisuutta. Tietojärjestelmän synonyyminä näkee joskus käytettävän termiä ohjelmisto tai ohjelmistojärjestelmä, mutta tällöin on kyse suppeammasta käsitteestä. Tietojärjestelmään voi nimittäin sisältyä yksi tai useampia ohjelmistoja, jotka suorittavat omia tehtäviään yhteisen päämäärän tavoittamiseksi.

Tietojärjestelmä voi olla automaattinen tai manuaalinen, tai se voi olla yhdistelmä edellisistä [Poh05, 338]. Automaattinen tietojärjestelmä koostuu tietoteknisistä komponenteista, jotka suorittavat itsenäisesti niille määritellyjä tehtäviä. Esimerkiksi ajastetut eräajojärjestelmät tai monet sulautetut järjestelmät ovat tyypillisiä automaattisia tietojärjestelmiä. Manuaalinen tietojärjestelmä puolestaan perustuu käyttäjän omaan työpanokseen. Tästä voidaan mainita esimerkkinä kirjanpitolvihkosta ja kuiteista koostuva käsin ylläpidettävä kirjanpitojärjestelmä. Yleensä tietojärjestelmässä yhdistyvät sekä automaattiset että manuaaliset, toisiaan täydentävät toimenpiteet. Tällöin keskeisessä asemassa ovat rajapinnat, joiden kautta syötteet (engl. input) ja tulosteet (engl. output) kul-



kevat. Tässä tutkielmassa termillä tietojärjestelmä viitataan kokonaisuuteen, joka voi sisältää sekä automaattisia että manuaalisia piirteitä. Tällä halutaan korostaa sitä, että tietojärjestelmä sisältää aina myös inhimillisen, käyttäjiin liittyvän puolen kuten vastuualueet, käyttäjäroolit ja toimintatavat. Tutkielmassa käytetään myös termiä järjestelmä tietojärjestelmän synonyyminä.

Tietojärjestelmiä voidaan luokitella eri tavoin pohjautuen esimerkiksi tavoitteisiin, toimialaan tai toteutukseen. Pohjonen [Poh05, 338-339] luokittelee tietojärjestelmät viiteen eri luokkaan niiden käyttötarkoituksen mukaan. *Henkilökohtaisilla tietojärjestelmillä ja toimistoautomaatiojärjestelmillä* suoritetaan päivittäisiä työtehtäviä, kuten viestintää tai dokumenttien hallintaa. Tällaisia tietojärjestelmiä ovat esimerkiksi sähköposti- ja tekstinkäsittelyohjelmistot. *Tapahtumankäsittelyjärjestelmät* on kehitetty ensisijaisesti kaupallishallinnollisiin tarpeisiin, ja niiden avulla suoritetaan esimerkiksi tilitapahtumien käsittelyä, rekistereiden ylläpitoa tai varastonhallintaa. *Reaaliaikaiset järjestelmät* puolestaan suorittavat muun muassa valvonta- ja prosessinohjaustehtäviä esimerkiksi sulautetuissa järjestelmissä kontrolloimalla ympäristöään, analysoimalla keräämäänsä dataa ja muokkaamalla tulosten perusteella tarvittaessa toimintaansa. *Päätöksentekijärjestelmät* analysoivat ja tuottavat tietoa ja ennusteita esimerkiksi avustamaan organisaatiojohtoa päätöksenteossa. *Asiantuntijajärjestelmät* puolestaan sisältävät johonkin suppeaan aihepiiriin sisältyvää tietoa ja avustavat kyseisen alan asiantuntijoita päätöksenteossa.

Tietojärjestelmä ei synny itsestään, vaan se tuotetaan tietoisesti vastaamaan johonkin tiettyyn tarpeeseen [Poh05, 340]. Yleensä taustalla on jokin asiakkaan esittämä ongelma, joka tuotettavan tietojärjestelmän on tarkoitus ratkaista. Kyseessä voi olla esimerkiksi jokin työläs manuaalinen toiminto, jota halutaan helpottaa tai nopeuttaa uuden tietojärjestelmän avulla. Joskus ongelmaksi on voinut muodostua jo käytössä oleva mutta vanhentunut tietojärjestelmä, joka ei enää palvele riittävällä tasolla. Tällöin ei välttämättä tarvita uutta järjestelmää, vaan keskitytään parantamaan ja tehostamaan vanhaa järjestelmää, eli tehdään niin sanottu tietojärjestelmäpäivitys. Tietojärjestelmän tuottamista ja päivittämistä kutsutaan tietojärjestelmien kehittämiseksi, jossa kehitystoimenpiteet voivat kohdistua niin ohjelmistoihin, laitteistoihin, prosesseihin, ohjeistuksiin kuin käyttäjiinkin.

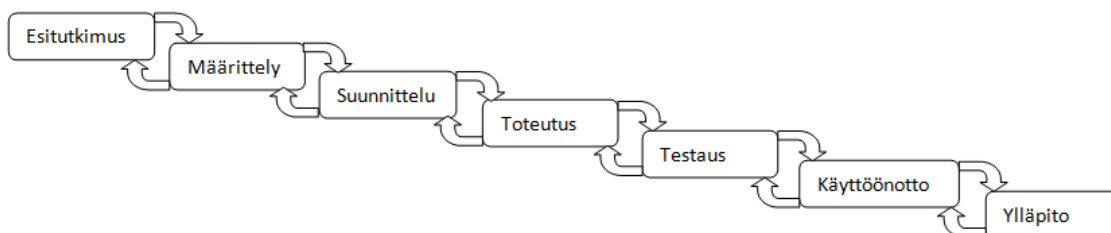
Koska tietojärjestelmän kehittäminen on melko työläs ja monimutkainen prosessi, se edellyttää suunnitelmallisuutta. Tämän vuoksi tietojärjestelmien kehitystyö organisoidaan yleensä ennalta määriteltyjen vaiheiden, tehtävien, tuotosten ja muiden elementtien avulla. Ne mahdollistavat tietojärjestelmän rakentamisen suunnitelmallisesti ja systemaattisesti tietyn kaavan mukaan. Tällöin puhutaan systeemityöstä ja ohjelmistoprosessista [Poh05, 340, 343]. Tietojärjestelmien kehittämiselle ominaista on vaihejako. Työ jakautuu vaiheisiin, jotka määrittävät kehitystyön aikaiset tehtäväkokonaisuudet ja vaiheistavat koko kehitysprojektin.

Yleisesti käytössä olevia vaiheita on Pohjosen [Poh05, 344-346] mukaan seitsemän. Ensimmäinen vaihe on *esitutkimus*, jossa selvitetään, onko kehityshanke ylipäänsä mahdollista ja mielekästä toteuttaa. Tämän jälkeen seuraa *määrittelyvaihe*, jossa kuvataan, mitä järjestelmän halutaan tekevän. Vaiheessa laaditaan järjestelmän toiminnallisen määrittely pohjautuen muun muassa asiakasvaatimuksiin. Määrittelyiden jälkeen tulee *suunnitteluvaihe*. Siinä toiminnallinen määrittely muunnetaan tekniseksi määrittelyksi eli laaditaan tarkat suunnitelmat, kuinka järjestelmä tullaan toteuttamaan. Tämän jälkeen on vuorossa *toteutusvaihe*, jossa ohjelmisto toteutetaan jollakin sovelluskehittimellä tai ohjelmointikielellä. Seuraavaksi toteutettu ohjelmisto *testataan* eli todennetaan, toimiiko se määrittelyiden ja suunnitelmien mukaisesti. Kun testaus on suoritettu hyväksytysti, voidaan siirtyä *käyttöönottovaiheeseen*. Siinä järjestelmä asennetaan tuotantoympäristöön ja suoritetaan tarvittavat migraatiot tiedoille ja tiedostoille. Lisäksi tässä vaiheessa tulee huolehtia käyttäjien ja ylläpitohenkilöiden koulutuksesta. Tämän jälkeen seuraa järjestelmän viimeinen ja pisin vaihe, *ylläpitovaihe*. Sen aikana huolehditaan järjestelmän toimivuudesta korjaamalla mahdollisia virheitä tai tekemällä muutoksia, ja lopulta poistetaan järjestelmä käytöstä sen elinkaaren päätteeksi. Vaiheet siis kattavat tietojärjestelmän elinkaaren alusta loppuun saakka.

Kumar ja Bhatia [KuB14, 189] esittävät, että vaiheiden sisällöt ja tehtävät ovat melko samanlaisia projektista tai organisaatiosta riippumatta. Sen sijaan niiden keskinäinen järjestys ja ajoitus voivat vaihdella suurestikin riippuen siitä, mikä vaihejakomalli eli systeemityömenetelmä projektissa on otettu käyttöön. Erilaisia systeemityömenetelmiä ovat esimerkiksi perinteinen vesiputousmalli, spiraalimalli, RUP ja modernit ketterät menetelmät. Seuraavaksi esitellään hieman tarkemmin kaksi ääripäätä, vesiputousmalli sekä ketteriä menetelmiä edustava Scrum. Näihin kahteen menetelmään tullaan palaa-

maan tässä tutkielmassa myöhemmin ja esittelemään, miten sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittelyä voitaisiin suorittaa osana näitä malleja.

Jo 1960-luvulla Roycen kehittämässä vesiputousmallissa [Roy70] työnkulku etenee kuvan 1 mukaisesti vaiheesta toiseen määrättyssä järjestyksessä. Uusi vaihe aloitetaan vasta edellisen vaiheen päätyttyä, ja taaksepäin palaamista pyritään välttämään hiomalla kukin vaihe kerralla loppuun saakka.

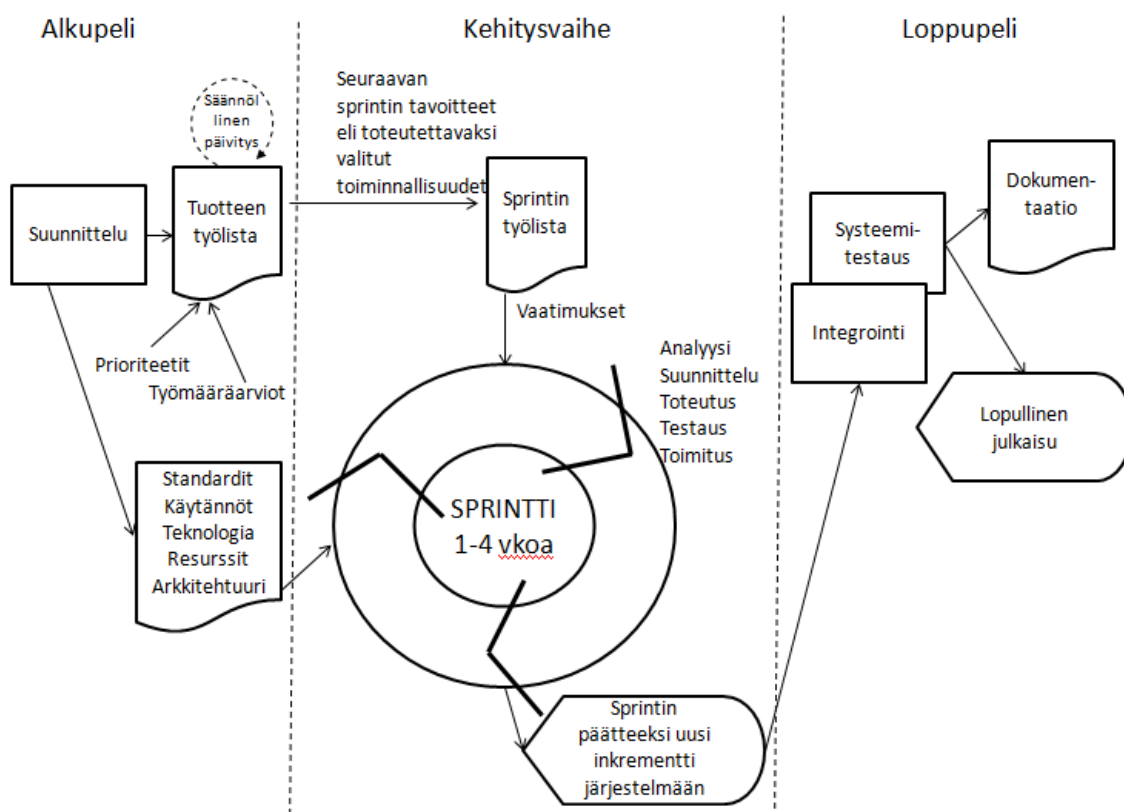


Kuva 1. Vesiputousmallin eteneminen [Roy70 mukaillen].

Roycen [Roy70] mukaan paluu on tarvittaessa kuitenkin mahdollista, mutta silloin joudutaan käytännössä suorittamaan uudelleen koko vesiputous paluukohdasta eteenpäin. Esimerkiksi vaatimukset hiotaan määrittelyvaiheessa kertaalleen valmiiksi ja niiden muuttamista vältetään. Jos sitten vaikkapa testausvaiheessa todetaan, ettei järjestelmä täytäkään tilaajan toiveita, joudutaan palaamaan takaisin määrittelyvaiheeseen täydentämään määrityksiä. Sen jälkeen tehdään uusien määritysten mukaiset suunnitelmat, toteutetaan järjestelmä uudelleen niiden mukaisesti, ja lopulta testataan jälleen, toimiiko järjestelmä nyt tilaajan toiveiden mukaisesti. Työtä joudutaan siis tekemään useaan kertaan, mikä vie aikaa ja kuluttaa resursseja. Vesiputousmalli on kuitenkin jäykkyydestään huolimatta edelleen hyvin laajasti käytössä oleva systeemityömenetelmä.

Moderneja systeemityömenetelmiä edustaa ketteriin menetelmiin kuuluva Scrum. Scrum on viitekehys, jossa määritellään kehitysprojektiin liittyvät roolit sekä ohjeistetaan kehitysprosessi, tuotokset ja työskentelytavat [ScS14, 3-4]. Kehitystyö etenee iteraatiivisesti ja inkrementaalisesti eli toistavasti ja lisäyksissä. Kuten kuvasta 2 ilmenee, työskentely jakautuu kolmeen vaiheeseen: alkupeliin (engl. pre game phase), kehitysvaiheeseen (engl. development phase) ja jälkipeliin (engl. post game phase) [ASR02, 27-35]. Vaiheita toistetaan kehitysprojektin aikana niin kauan, että päästään toivottuun lopputulokseen. Alkupelin päätehtävä on suunnittelu, jonka puitteissa tehdään esimerkiksi arkkitehtuuriin, resursseihin ja työkaluihin liittyvät suunnitelmat. Lisäksi alkupeli-

vaiheessa kootaan ja ylläpidetään tuotteen työlistaa (engl. product backlog). Työlistalla on priorisoituna kaikki tulevaan järjestelmään toivottavat ominaisuudet ja toiminnallisuudet sekä niihin liittyvät työmääräarviot. Tuotteen työlistalta valitaan korkeimman prioriteetin ominaisuudet seuraavan sprintin työlistalle (engl. sprint backlog). Kehitysvaiheessa sprintin työlistalta sitten otetaan työn alle tehtäviä niiden prioriteettijärjestyksessä.



Kuva 2. Scrum-mallia noudattavan kehitystyön eteneminen [ASR02, 28 mukaillen].

Varsinainen työskentely tapahtuu lyhyissä, noin kuukauden mittaisissa sprinteissä [ScS14, 7-14]. Niiden aikana suoritetaan vaatimusten analysointia, laaditaan ja tarkennetaan suunnitelmia, toteutetaan tietojärjestelmän osia ja suoritetaan niiden testausta. Sprintin lopuksi tuotokset katselmoidaan ja toimitetaan ennalta sovittu inkrementti kehityksen kohteena olevaan järjestelmään. Lopuksi jälkipelivaiheessa suoritetaan hyväksytyn inkrementin integrointi muihin valmiisiin järjestelmäosiin ja suoritetaan systeemi-testaus ennen lopullista julkaisua. Samalla viimeistellään julkaisuun liittyvä dokumentaatio. Ennen uuden sprintin aloitusta tarkastellaan mennyttä sprinttiä ja pohditaan, onko jotakin käytäntöjä tai menettelyitä tarpeen muuttaa. Syntyneitä oivalluksia hyödynne-

tään seuraavaksi alkavassa alkupelivaiheessa, jossa suunnitellaan seuraava sprintti ja sykli jatkuu, kunnes koko järjestelmä on valmis ja hyväksytty tuotantoon.

Asiakas tai tilaaja sitoutetaan mukaan kehitystyöhön tuotteen omistajan (engl. product owner) roolissa [ScS14, 5]. Hänen tehtävänä on vastata tuotteen työlistasta ja sen priorisoinnista sekä katselmoida ja hyväksyä sprinttien tuotokset. Näin saadaan säännöllisesti palautetta siitä, eteneekö työ oikeaan suuntaan vai tuleeko vaatimuksia muuttaa. Vaatimusten siis oletetaan elävän projektin aikana, ja niitä voidaan tarkentaa tai muuttaa aina seuraavaa sprinttiä varten. Menettely parantaa kehitystyön ennakoitavuutta ja pienentää siihen liittyviä riskejä.

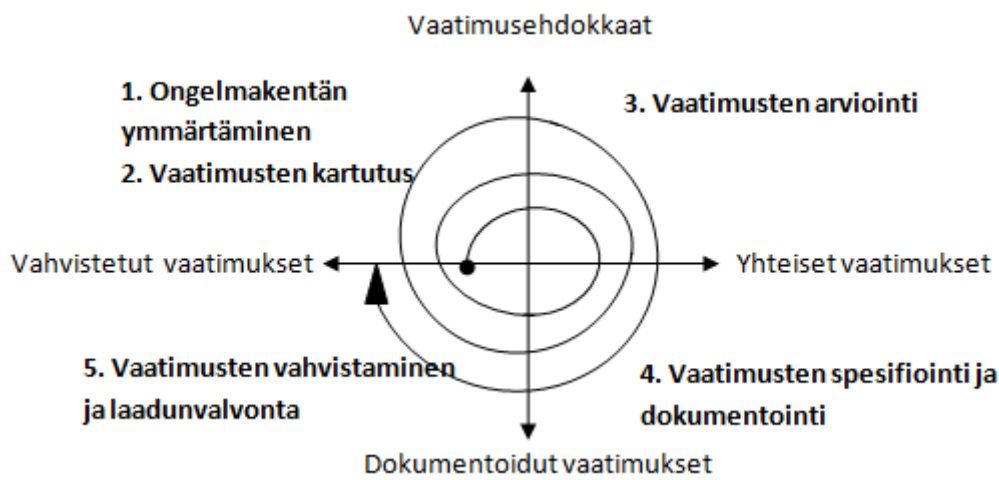
Jokaisella systeemityömenetelmällä on siis omat prosessinsa ja ohjeistuksensa kehitystyön edistämiseen. Käytännössä menetelmät ovat lähinnä suuntaa-antavia kuvauksia siitä, miten kehitystyö voisi ideaalitapauksessa edetä. Hyvin harvoin niitä pystytään noudattamaan tai olisi edes tarkoituksenmukaista noudattaa kirjaimellisesti. Kumar ja Bhatia [KuB14, 196] tuovat esille myös sen, ettei yksi malli sovi kaikkiin projekteihin. Esitutkimuksen yhteydessä onkin järkevää pohtia, mikä malli soveltuisi kyseiseen kehitysprojektiin parhaiten.

Tässä tutkielmassa tullaan perehtymään systeemityön vaiheista erityisesti vaatimusmäärittelyyn. Muita vaiheita käsitellään vain siinä määrin, kun se on vaatimusmäärittelyn kannalta olennaista. Keskittyminen nimenomaan vaatimusmäärittelyyn on perusteltua, sillä vaatimusmäärittelyn voidaan sanoa olevan tietojärjestelmäkehityksen perustelu ja selkäranka. Jos vaatimuksia ei ole selvitetty ja dokumentoitu huolella, joutuvat toteuttajat arvailemaan asiakkaan tarpeita, ja projektin onnistuminen on kiinni onnesta. Puuttuvien ja virheellisesti määriteltyjen vaatimusten onkin todettu olevan yksi merkittävimmistä syistä kehitysprojektien epäonnistumiseen [The95, 4-5 ja FGZ15, 26]. Vastaavasti huolella tehty vaatimusmäärittely tukee muita kehitysprosessin vaiheita nopeuttaen kehitystyön etenemistä, alentaen kustannuksia ja kasvattaen asiakastyytyväisyyttä [AtM11, 183-184].

## **2.2 Vaatimusmäärittelyprosessin työvaiheet**

Van Lamsweerden mukaan vaatimusmäärittelyprosessiin sisältyy viisi työvaihetta, jotka toistuvat melko samantyyppisinä valitusta systeemityömenetelmästä riippumatta

[Lam09, 34-35]. Vaiheet ovat seuraavat: ongelmakentän ymmärtäminen (engl. domain understanding), vaatimusten kartutus (engl. elicitation), vaatimusten arviointi (engl. evaluation), vaatimusten spesifiointi ja dokumentointi (engl. specification and documentation) sekä vaatimusten vahvistaminen (engl. consolidation). Kuten kuvasta 3 ilmenee, vaiheet seuraavat toisiaan iteratiivisessa syklissä, jonka myötä vaatimukset tarkentuvat asteittain. Jokaisesta vaiheesta on tuloksena jalostuneempi käsitys vaatimuksista, ja kunkin vaiheen tulokset toimivat lähteenä seuraavalle vaiheelle.



Kuva 3. Vaatimusmäärittelyprosessin työvaiheet [Lam09, 34 mukaillen].

Ensimmäinen työvaihe on kehityskohteen *ongelmakentän ymmärtäminen* [Lam09, 30]. Ymmärrystä saadaan selvittämällä ja analysoimalla kehitysprojektissa tuotettavan järjestelmän avulla ratkaistava ongelma sekä siihen liittyvät yhteydet ja reunaehdot. Yleensä lähdetään liikkeelle analysoimalla nykyratkaisua (engl. the system-as-is), jotta saadaan selville, mitä asioita, tahoja ja toimintoja kokonaisuuteen tällä hetkellä liittyy ja miten kokonaisuus nyt toimii [Lam09, 3-6]. Nykyratkaisun pohjalta pystytään hahmottelemaan tulevaa ratkaisua (engl. the system-to-be), jossa toiminta pohjautuu kehitysprojektin kohteena olevaan tietojärjestelmään. Vaatimusmäärittelyä voidaan viedä tätäkin pidemmälle, jos siinä huomioidaan myös tulevia tarpeita. Tällöin pystytään ennakkoimaan järjestelmän kehitystä ja versiointia eli tulevaisuuden ratkaisua (engl. the systems-to-be-next).

Ongelmakentän ymmärtämisessä voidaan käyttää apuna WHY-WHAT-WHO -kysymyspatteristoa [Lam09, 12-16]. Sen mukaisesti on ensin kiteytettävä itse ongelma etsimällä vastauksia kysymyksiin "miksi nykyjärjestelmä ei toimi, eli mikä on ratkaista-

va ongelma" (WHY). Vastauksista saadaan tietoa toimialasta sekä tavoitteista, joihin kehitystyössä pyritään. Seuraavaksi hahmotellaan odotukset uudelle järjestelmälle kysymällä: "mitä uuden järjestelmän pitäisi tehdä, eli miten ongelma ratkaistaan" (WHAT). Vastausten pohjalta saadaan tietoon uuteen järjestelmään tarvittavat toiminnallisuudet ja ominaisuudet. Lopuksi selvitetään, mitä toimijoita ja vastuutahoja ongelman ratkaisuun liittyy kysymällä: "ketkä tai mitkä vastaavat ongelman ratkaisusta" (WHO). Näin saadaan ymmärrystä siitä, mitä laitteita, ohjelmistoja, henkilöitä tai tahoja tuleva ratkaisu tulee sisältämään. Samalla hahmottuvat vaatimusmäärittelyn sidosryhmät eli tahot, joiden on tarpeen osallistua vaatimusmäärittelyprosessiin.

Edellä esitellyn kysymyspatteriston ohella ymmärrystä ongelmakentästä hankitaan muillakin keinoilla [Lam09, 31]. Vaiheen aikana kannattaa tutkia olemassa olevaa dokumentaatiota, tarkastella nykyistä tai muita vastaavanlaisia järjestelmiä sekä haastatella tai havainnoida järjestelmän käyttöön ja hallintaan liittyviä henkilöitä. Näin saadaan koottua tietoa eri lähteistä ja näkökulmista käsin, ja vähitellen ongelmakenttä alkaa hahmottua.

Työvaiheen toimien tulokset dokumentoidaan huolellisesti [Lam09, 31]. Vaiheen tuotoksena on alustava vaatimusdokumentti, jonka pohjana voi käyttää esimerkiksi jotakin Internetistä saatavaa mallipohjaa. Lisäksi vaiheen aikana aloitetaan koko prosessin aikana käytettävän sanaston kokoaminen. Sanastoon kootaan yhteinen ymmärrys järjestelmään liittyvistä termeistä ja niiden määritelmistä. Yhteisen sanaston avulla varmistetaan, että kaikki vaatimusmäärittelyn osapuolet ymmärtävät asiat samalla tavalla. Cysneiros [Cys02] tuo esille sanaston hyötynä myös ajansäästön. Kun kehitysprojekteissa on ollut käytössä sanasto, on uusilla projektin jäsenillä kulunut alueeseen perehtymiseen aikaa vain kolmannes siitä, mitä vastaavissa projekteissa ilman sanastoa on kulunut.

Ongelmakentän ymmärtämistä seuraava työvaihe on *vaatimusten kartutus* [Lam09, 31-32]. Sen tarkoituksena on löytää tulevaan ratkaisuun liittyviä vaatimusehdokkaita sekä oletuksia ja kuvata näihin pohjautuvia vaihtoehtoisia ratkaisuja. Käytännössä vaatimusten kartutusta tehdään usein samalla, kun pyritään saamaan ymmärrystä ongelmakentästä, eli vaiheita ei ole välttämätöntä erottaa toisistaan. Kartutuksen tukena käytetään siihen mennessä kertynyttä tietoa ongelmakentästä, esimerkiksi nykyratkaisussa tunnistettuja puutteita ja ongelmia. Lisäksi pyritään huomioimaan teknologian tai markkinatilan-

teen tarjoamat mahdollisuudet, jotka voivat tarjota uusia käyttötapoja tai ominaisuuksia. Toisaalta myös mahdolliset teknologiset ja organisaation asettamat rajoitukset on huomioitava, jotta vaatimukset olisivat realistisia. Tyypillisiä tapoja kartuttaa vaatimuksia ovat haastattelut, havainnointi sekä prototyyppien tekeminen ja niiden arviointi. Vaiheen aikana voidaan myös mallintaa tyypillisimmät skenaariot, joissa kuvataan tulevan ratkaisun vuorovaikutusta ympäristönsä kanssa. Skenaariot auttavat ymmärtämään, miten uusi järjestelmä tulee toimimaan ja mitä asioita on otettava huomioon.

Kartutusvaiheen tuotoksena on edellisestä vaiheesta tarkennettu alustava vaatimusdokumentti, johon on koottu esille tulleet vaatimusehdokkaat ja rajoitukset sekä mahdollisia ratkaisuvaihtoehtoja [Lam09, 32]. On kuitenkin huomioitava, että vaatimusten kartutus on luonteeltaan yhteinen oppimisprosessi. Ei ole edes tarkoitus, että heti ensimmäisellä kierroksella osattaisiin ilmaista täydelliset, lopulliset vaatimukset. Sen sijaan lähdetään yleensä liikkeelle karkeamman tason kokonaisuuksista, ja vasta ajan myötä niistä tarkennetaan iteratiivisesti yksityiskohtaisempia vaatimuksia. Van Lamsweerde korostaa, että kartutus on erittäin kriittinen työvaihe. Jos kartutusta ei ehditä tai osata suorittaa huolellisesti, on tuloksena huonoja vaatimuksia, joiden seurauksena kehitettävä järjestelmä ei toimi toivotulla tavalla ja siinä on merkittäviä puutteita.

Seuraava työvaihe on *vaatimusten arviointi* [Lam09, 32-33]. Vaiheen tavoitteena on luoda kaikkien vaatimusmäärittelyn osapuolten yhteinen, hyväksytty näkemys vaatimuksista, jotka tulevan ratkaisun tulee täyttää. Tämä edellyttää edellisessä vaiheessa esille tulleiden ristiriitojen ratkaisemista neuvottelemalla sekä tulevan järjestelmän riskien huolellista analysoimista. Samalla edellisessä vaiheessa esitettyjä ratkaisuvaihtoehtoja arvioidaan ja niistä valitaan tämänhetkisen ymmärryksen mukaan paras vaihtoehto. Lisäksi tunnistetut vaatimukset priorisoidaan eli luodaan yhteinen käsitys vaatimusten keskinäisestä tärkeysjärjestyksestä. Näin saadaan tukea toimitusaikataulun laatimiseen; mihin ominaisuuksiin kehitystyössä tulee ensisijaisesti panostaa ja mitkä ominaisuudet voidaan mahdollisesti jättää julkaistavaksi myöhemmin.

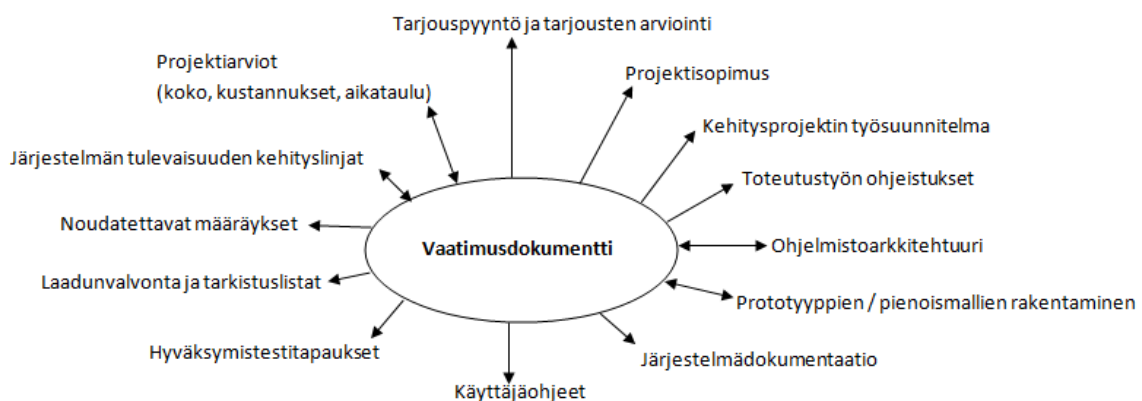
Vaatimusten arvioinnin tukena voidaan käyttää erilaisia kvalitatiivisia ja kvantitatiivisia menetelmiä, kuten virhepuut (engl. fault tree) tai todennäköisyyslaskelmat [Lam09, 33, 96-99]. Vaiheen tuloksena on edellisissä vaiheissa alustetun dokumentin päivitetty ver-



sio, jonne on koottu arviointivaiheessa kiinnitetyt ja priorisoidut, yhteiset vaatimukset sekä tulevaan ratkaisuun liittyvät oletukset.

Kun vaatimukset on arvioitu, siirrytään seuraavaan työvaiheeseen: *vaatimusten spesifiointiin ja dokumentointiin* [Lam09, 33]. Vaiheen tarkoitus on tarkentaa ja dokumentoida edellisessä vaiheessa muotoillut vaatimukset ja niihin liittyvät järjestelmäominaisuudet. Tarvittaessa vaatimukset voidaan dokumentoida jollakin formaalilla spesifikaatiokielellä ja tukena voidaan käyttää erilaisia kaavioita. Tuloksena on varsinainen vaatimusdokumentti (engl. RD eli Requirements Document), johon on kirjattu tulevan ratkaisun tavoitteet, määritelmät, vastuut, olennaiset ongelmakentän ominaisuudet, käyttöympäristöön liittyvät oletukset sekä järjestelmä- ja ohjelmistovaatimukset. Nämä muodostavat dokumentin ytimen, mutta tarvittaessa dokumenttiin voidaan sisällyttää myös esimerkiksi perusteluita tehdyille valinnoille, kustannusarvioita, hyväksymistestaukseen liittyvää tietoa tai toimialan teknisiä liitteitä.

Vaiheessa koottua vaatimusdokumenttia hyödynnetään ainakin jossain määrin kaikissa järjestelmän elinkaarivaiheissa [Lam09, 42-43]. Kuvassa 4 esitellään toiminnot, joihin vaatimusdokumentilla on vaikutusta. Dokumentista saadaan tietoa kehitysprojektin tulevasta laajuudesta projektin suunnittelun tueksi sekä yksityiskohtia mahdollisia kilpailutuksia tai tarjouspyyntöjä varten. Sen avulla pystytään myös tekemään projektisopimukset sekä työsuunnitelmat ja saadaan ohjeistuksia toteutustyötä varten. Vaatimusdokumenttiin kirjattuja vaatimuksia ja hyväksymistestitapauksia hyödynnetään testauksessa ja laadunvarmistuksessa.



Kuva 4. Vaatimusdokumentin käyttö kehitysprojektin aikana [Lam09, 43 mukaillen].

Kuten kuvasta 4 ilmenee, vaatimusdokumenttia käytetään tukena järjestelmädokumentaation sekä käyttäjäohjeistuksien laatimisessa. Se myös kokoaa järjestelmän käyttöön liittyvät määräykset. Järjestelmän ylläpitovaiheessa dokumentin sisältämät yksityiskohdat toimivat luontevana tarkistuslistana, kun järjestelmän toimintaa ja laatua valvotaan. Kaksisuuntaiset nuolet kuvaavat tilannetta, jossa vaikutusta on molempiin suuntiin, eli toiminto vaikuttaa omalta osaltaan vaatimusdokumenttiin. Tällaisia kaksisuuntaisia toimintoja ovat projektiarvioiden laatiminen, ohjelmistoarkkitehtuurityö, prototyypityskentely sekä järjestelmän tulevien versioiden hahmottelu. Niistä kaikista saadaan tietoon lisävaatimuksia tai rajoitteita, jotka on päivitettävä vaatimusdokumenttiin.

Kuvan 3 mukaisen vaatimusmäärittelysyklin viimeinen vaihe on *vaatimusten vahvistaminen ja laadunvalvonta* [Lam09, 33, 187-188]. Vaiheen aikana pyritään varmistamaan, että siihen mennessä dokumentoidut vaatimukset ovat valideja eli täyttävät sidosryhmien todelliset tarpeet. Samalla vaatimukset verifioidaan eli niitä tarkastellaan suhteessa toisiinsa mahdollisten epäjohtonmukaisuuksien ja puutteiden havaitsemiseksi. Jos validoinnissa tai verifiointissa havaitaan ongelmia, ne tulee ratkaista, ennen kuin vaatimusdokumentti toimitetaan suunnittelijoille ja toteuttajille. Tarvittaessa tunnistettuihin ongelma-kohtiin voidaan palata vaatimusmäärittelyprosessin seuraavassa syklissä, kun vaatimuksia tarkennetaan iteratiivisesti.

Van Lamsweerde [Lam09, 33] muistuttaa, mitä myöhemmin vaatimukseen liittyvä ongelma havaitaan, sitä kalliimmaksi sen korjaaminen tulee. Laadunvalvonnan tekniikoita ovat esimerkiksi tarkistukset (engl. inspections), katselmoinnit (engl. reviews) ja läpikäynnit (engl. walk-throughs). Jos vaatimukset on tallennettu määriteltyjen notaatioiden avulla tietokantaan, voidaan laadunvarmistuksen tukena tehdä vaatimusten rakenteen johtonmukaisuutta ja kuvausten kattavuutta selvittäviä kyselyitä. Lisäksi laadunvalvonnan tukena voidaan tarvittaessa käyttää apuna erilaisia tarkistuslistoja. Van Lamsweerde [Lam09, 191-192] on koonnut tarkistuslistan, jonka avulla dokumentaatiosta voidaan etsiä esimerkiksi ristiriitoihin, soveltuvuuteen, mitattavuuteen ja ymmärrettävyyteen liittyviä virheitä. Vaatimusmäärittelyn laatuun palataan tarkemmin luvussa 2.4.

Vaiheen tuloksena on aikaisempien vaiheiden tuotoksiin pohjautuva dokumentti vahvistetuista vaatimuksista [Lam09, 33]. Kaikki mahdolliset virheet ja ongelmat tulee olla korjattuna myös aiemmissa vaiheissa tehtyihin dokumentteihin ja kirjauksiin, jotta ne

eivät aiheuttaisi väärinkäsityksiä myöhemmin. Päivitetyn vaatimusdokumentin ohella muita vaiheen tuotoksia voivat olla esimerkiksi vaatimusten validoinnissa apuna käytettävä prototyyppi tai pienoismalli, verifiointin tuloksena tarkentuneet hyväksymistestitapaukset, alustava toteutussuunnitelma tai alihankkijoille lähetettävä tarjouspyyntö.

Kehitysoikeudessa valitusta systeemityömenetelmästä riippuu, milloin vaatimusmäärittelyä tehdään. Ääripäinä voidaan nähdä luvussa 2.1 esitelty perinteisiä malleja edustava vesiputousmalli sekä moderneja ketteriä malleja edustava Scrum [KuB14, 190, 194-195]. Vesiputousmallissa vaatimusmäärittely suoritetaan kertaluonteisesti yhtenä vaiheena projektin alussa, jolloin vaatimukset lopputuotteelle kiinnitetään hyvin varhaisessa vaiheessa. Tällöin kuvan 3 mukaista sykliä toistetaan niin monta kertaa, että vaatimukset ovat selvillä ja niistä voidaan laatia tekniset suunnitelmat toteutusta varten. Projektin lopussa valmis tuotos esitellään asiakkaalle, ja vasta sitten voidaan arvioida, olivatko vaatimukset oikein ja riittävän kattavasti määriteltäviä. Jos asiakas ei olekaan tyytyväinen tai muuttaa mieltään, on muutosprosessi hidas ja kallis, sillä koko vesiputous joudutaan käymään läpi uudelleen vaihe vaiheelta.

Scrumissa sen sijaan vaatimusmäärittelyä tehdään käytännössä koko projektin ajan [KuB14]. Tämä juontaa juurensa ketterien menetelmien taustalla oleviin periaatteisiin, joihin on kirjattu, että muuttuvat vaatimukset otetaan vastaan myös kehityksen myöhemmissä vaiheissa. Kuvan 3 sykli voidaan siten toistaa aina tarvittaessa, kun on saatu lisäymmärrystä tulevasta järjestelmästä esimerkiksi alustavien toteutuksien pohjalta. Tällaisella menettelyllä pystytään tarvittaessa reagoimaan nopeasti vaatimusten muutoksiin ja varmistamaan, että lopputuotos tyydyttää asiakasta.

Tapahtui vaatimusmäärittely missä vaiheessa projektia tahansa, vaatimukset ovat läsnä koko projektin ajan [Lam09, 42-43 ja AtM11, 183-184]. Ne toimivat pohjana teknisille suunnitelmille ja toteutukselle, ja niistä muodostetaan testitapaukset, joita vasten testataan lopulta järjestelmän toimivuus. Kehitysoikeudessa tulee myös tiedostaa, että vaatimusmäärittelyssä tulisi kartoittaa nykytarpeiden ohella myös tulevaisuuden tarpeita. Tällöin voidaan tehdä ratkaisuja, joiden avulla järjestelmä pystyy mukautumaan toimialan tai reunaehtoien muutoksiin kohtuullisen helposti. Jos huomio kiinnitetään ainoastaan määrittelyhetkellä akuutteihin tarpeisiin ja vaatimuksiin, on olemassa suuri riski, ettei järjestelmä pysty vastaamaan lähitulevaisuuden tarpeisiin ja haasteisiin. Tällöin

organisaation on käynnistettävä järjestelmän uudistusprojekti, tai pahimmassa tapauksessa järjestelmä on lakkautettava vanhentuneena ja hankittava tilalle uusi järjestelmä. Tällainen lyhytnäköisyys tuottaa organisaatiolle turhia menoeriä ja kuluttaa resursseja.

### 2.3 *Vaatimusmäärittelyssä tunnistettujen vaatimusten luokitteluja*

Vaatimusmäärittelyprosessin tavoitteena on tunnistaa tulevaan järjestelmään kohdistuvat vaatimukset. Vaatimuksia voidaan luokitella eri tavoin riippuen näkökulmasta, jota halutaan korostaa. Yleinen jako on vaatimusten luokittelu järjestelmä- ja ohjelmistovaatimuksiin [Lam09, 18-19]. Van Lamsweerden mukaan järjestelmävaatimuksilla tarkoitetaan koko tietojärjestelmään ja sen toimintaympäristöön liittyviä vaatimuksia, joita kehitettävä ohjelmisto toteuttaa yhdessä muiden järjestelmän komponenttien kanssa. Ohjelmistovaatimukset puolestaan kohdistuvat ainoastaan kehitettävään ohjelmistoon tai siihen liittyviin rajapintoihin. Ohjelmistovaatimukset toimivat lähtökohtana ohjelmiston kehitystyölle. On erittäin tärkeää, että ohjelmistovaatimukset on kirjattu sellaisella tavalla, että toteuttajat ymmärtävät ne yksiselitteisesti.

Vaatimukset voidaan jakaa myös toiminnallisiin (engl. functional requirements) ja ei-toiminnallisiin vaatimuksiin (engl. non-functional requirements) [Lam09, 23-24]. Seuraavaksi tutustutaan van Lamsweerden esittämiin vaatimusten luokitteluihin hieman tarkemmin käyttäen esimerkkinä hypoteettista insuliinipumpun toimintajärjestelmää. Vaatimusesimerkit ovat täysin mielikuvituksellisia, ja niillä on tarkoitus ainoastaan selvittää erilaisia vaatimusluokkia.

Van Lamsweerden [Lam09, 23] mukaan toiminnalliset vaatimukset kuvaavat niitä konkreettisia toiminnallisuuksia ja palveluita, joita tuleva järjestelmä tarjoaa käyttäjilleen. Niiden avulla määritellään, MITÄ järjestelmän tulee tehdä. Toiminnallisilla vaatimuksilla viitataan siis käytännössä luvussa 2.2 esitellyn WHY-WHAT-WHO -kysymyspatteriston WHAT-osuuteen sisältyviin toimintoihin. Esimerkkinä toiminnallisesta vaatimuksesta voidaan mainita insuliinipumpun toimintajärjestelmän vaatimus: *Insuliinia tulee voida annostella sekä automaattisesti että manuaalisesti*. Toimintojen ohella toiminnallisissa vaatimuksissa voidaan viitata myös toimintaympäristöön, joka voi asettaa toimintaehdoja tulevalle järjestelmälle. Tällainen ehto voisi olla esimerkiksi: *Ympäristön lämpötila ei saa vaikuttaa insuliinipumpun verensokerimittausten tuloksiin*. Toiminnallisten vaatimusten mallintamisessa voidaan käyttää apuna skenaarioita ja niis-

tä johdettuja käyttötapauksia, joilla kuvataan käyttäjien ja järjestelmän välisiä toimintoja.

Ei-toiminnalliset vaatimukset puolestaan kuvaavat rajoituksia ja reunaehtoja, joiden puitteissa tulevan järjestelmän tulee toimia [Lam09, 24]. Näkökulmana on se, MITEN järjestelmän tulee toimia tarjotessaan palveluita käyttäjälle. Cao ja Ramesh [CaR08, 64] varoittavat, että ei-toiminnallisten vaatimusten tunnistamisen on todettu olevan haasteellista erityisesti ketterissä kehitysprojekteissa. Niissä saatetaan keskittyä vaatimusmäärittelyssä pääasiassa järjestelmän toiminnallisiin, ja ei-toiminnalliset vaatimukset unohdetaan kokonaan. Jotta kehitysprojekteissa kiinnitettäisiin riittävästi huomiota myös ei-toiminnallisiin vaatimuksiin, on avuksi kehitetty erilaisia luokitteluja. Kuvassa 5 esitellään van Lamsweerden [Lam09, 24] laatima jaottelu, joka helpottaa ei-toiminnallisten vaatimusten tunnistamista eri näkökulmista käsin. Van Lamsweerden mukaan ei-toiminnalliset vaatimukset jaetaan neljään pääkategoriaan: laatu-, mukautuvuus-, arkkitehtuuri- ja kehitystyön vaatimuksiin. Lisäksi niillä voi olla tarkentavia alaluokkia.



Kuva 5. Ei-toiminnallisten vaatimusten luokittelu van Lamsweerden mukaan [Lam09, 24 mukaillen].

Van Lamsweerden [Lam09, 24] mukaan ensimmäinen pääluokka on laatuvaatimukset (engl. quality of service). Laatuvaatimuksilla tarkoitetaan laatuominaisuuksia, jotka määrittelevät, kuinka hyvin tulevan järjestelmän tulee toimia. Kuten kuvasta 5 ilmenee, laatuvaatimukseen sisältyy kuusi alaluokkaa: turvallisuus, suojaus, luotettavuus, suorituskyky, liittymät ja tarkkuus. Turvallisuudella (engl. safety) viitataan ominaisuuksiin,

jotka estävät järjestelmän toiminnan tilanteissa, joissa saattaisi olla seurauksena onnettomuuksia tai muita haittoja. Esimerkkinä käytetyn insuliinipumpun ohjausjärjestelmälle tällainen vaatimus voisi olla esimerkiksi: *Insuliinia voi annostella tunnin sisällä yhteensä maksimissaan 40 yksikköä riippumatta siitä, onko annostelu automaattista tai manuaalista.*

Suojaus (engl. security) puolestaan tarkoittaa tulevan järjestelmän osien tai resurssien suojaamista ulkoisilta tekijöiltä ja väärinkäytöksiltä [Lam09, 25]. Tämän luokan alaluokkia ovat luottamuksellisuus, eheys ja saatavuus. Luottamuksellisuudella (engl. confidentiality) viitataan siihen, että järjestelmän tietoja pääsevät katsomaan ja käyttämään ainoastaan siihen valtuutetut henkilöt. Tätä kuvaava vaatimus voisi olla esimerkiksi: *Insuliinipumpun suorittamien verensokerimittausten tuloksia voi lukea ainoastaan insuliinipumpun omistaja tai tietyn käyttäjäkoodin omaava terveydenhuoltohenkilöstö.* Eheys (engl. integrity) puolestaan tarkoittaa sitä, että järjestelmän tietojen tulee olla oikeellisia, ja niiden tuottaminen ja päivittäminen on sallittu ainoastaan valtuutettujen henkilöiden toimesta ennalta sovitulla menettelyillä. Eheysvaatimus voisi olla vaikkapa: *Insuliinipumpun suorittamia verensokerin mittaustuloksia ei voi muokata manuaalisesti.* Luottamuksellisuus- ja eheysvaatimuksissa siis kuvataan, mitkä tahot tietoja saavat käyttää. Saatavuusvaatimuksissa (engl. availability) taas määritellään, miten järjestelmän resurssit tai tiedot ovat valtuutettujen henkilöiden saatavilla. Esimerkki saatavuusvaatimuksesta voisi olla: *Insuliinipumpun suorittamien verensokerimittaustulosten tulee olla luettavissa 30 vuorokautta mittaushetkestä eteenpäin.*

Kolmas alaluokka laatuvaatimuksille on luotettavuus (engl. reliability) [Lam09, 25]. Luotettavuudella tarkoitetaan tässä yhteydessä sitä, miten hyvin tuleva järjestelmä toimii sen toimintaympäristössä esiintyvistä häiriöistä huolimatta ja miten hyvä sen vian-sietokyky on. Insuliinipumpun vaatimusmäärittelyssä tällainen vaatimus olisi esimerkiksi: *Jos insuliinipumpun verensokerimittaus epäonnistuu, täytyy käyttäjän siitä huolimatta voida annostella insuliinia manuaalisesti.*

Neljäs laatuvaatimusten alaluokka on suorituskky (engl. performance) [Lam09, 26]. Suorituskkyvaatimuksilla määritellään tulevan järjestelmän toimintaehtoja esimerkiksi vasteaikojen, kuormituksen tai tilan näkökulmasta. Kuten kuvassa 5 on esitetty, suorituskkyvaatimukset jakautuvat kolmeen alaluokkaan. Vasteaikavaatimus (engl. time)

ottaa kantaa siihen, miten kauan jonkin toiminnon suorittamiseen saa kulua aikaa tai miten nopeasti järjestelmän on tuotettava vastine käyttäjän toiminnolle. Insuliinipumpun toiminnassa vasteaikavaatimus voisi olla vaikkapa: *Insuliinipumpun on annosteltava insuliini käyttäjän elimistöön viiden sekunnin sisällä siitä, kun käyttäjä on aktivoinut manuaalisen annostelutoiminnon.*

Tilavaatimus (engl. space) puolestaan määrittelee sen, miten paljon muistia tai kovalevytilaa toiminnon suorittamiseen saa kulua. Tällainen esimerkki olisi: *Insuliinipumpun yksittäinen järjestelmään tallennettava verensokerimittaustulos saa viedä tilaa korkeintaan 10 kilotavua.* Kuluvaatimuksilla (engl. cost) määritellään, miten paljon kustannuksia tai työtä joku toiminto saa aiheuttaa tai miten paljon säästöjä uudesta järjestelmästä tulee saada nykytilanteeseen verrattuna. Esimerkiksi: *Insuliinipumpun suorittamien verensokerimittausten tulosten automaattisen synkronoinnin Omakanta-palveluun tulee tuottaa 75 prosentin aikasäästö verrattuna siihen, että tulokset syötettäisiin käsin,* olisi tällainen vaatimus.

Viides laatuvaatimusten alaluokka on liittymät eli rajapinnat (engl. interface) [Lam09, 26]. Liittymillä kuvataan sitä, miten tuleva järjestelmä on yhteydessä ympäristöönsä. Niissä voidaan ottaa kantaa staattisten ja dynaamisten yhteyksien syötteisiin ja tulosteisiin sekä vuorovaikutussekvensseihin. Kuvan 5 mukaisesti liittymillä on kolme alaluokkaa, jotka painottavat erityyppisiä yhteystahoja: käyttöliittymät, laitteistoliittymät sekä ohjelmistoliittymät. Käyttöliittymävaatimukset (engl. user interaction) vastaavat erityisesti ihmiskäyttäjien tarpeisiin, ja ne jakautuvat vielä kahteen alaluokkaan. Käytettävyyksivaatimuksilla (engl. usability) määritellään järjestelmän käyttöä, dialogeja, syötteitä ja tulosteita käyttäjien näkökulmasta. Insuliinipumpun käytettävyyksivaatimus voisi olla vaikkapa: *Kun käyttäjä yrittää annostella insuliinia manuaalisesti, on insuliinipumpun toimintajärjestelmän kysyttävä varmistus ja näytettävä aiottu annostelumäärä näytöllä.* Käyttömukavuusvaatimuksilla (engl. convenience) puolestaan otetaan kantaa siihen, millaisilla ominaisuuksilla varmistetaan, että käyttäjien on mahdollisimman mukavaa ja helppoa käyttää järjestelmää. Insuliinipumpun käyttömukavuusvaatimus voisi olla: *Käyttäjän on voitava säädellä insuliinipumpun näytöllä näkyvää fonttia kolmeen eri kokoluokkaan - pieni = 6pt, keskiuuri = 12pt ja suuri = 16pt.*

Liittymät-alaluokkaan kuuluvat käyttöliittymien ohella myös laitteistoliittymät sekä ohjelmistojen yhteentoimivuus [Lam09, 26]. Laitteistoliittymävaatimuksilla (engl. device interaction) määritellään sitä, miten tulevan järjestelmän ohjelmistot ovat vuorovaikutuksessa järjestelmän tai muun toimintaympäristön laitteisiin. Insuliinipumpun laitteistoliittymävaatimus voisi olla esimerkiksi: *Käyttäjän on voitava yhdistää insuliinipumppu tietokoneeseen esimerkiksi käyttöjärjestelmäpäivityksiä varten.* Ohjelmistojen yhteentoimivuudella (engl. software interoperability) puolestaan kuvataan sitä, miten järjestelmän ohjelmisto tai ohjelmistot ovat vuorovaikutuksessa keskenään tai muiden toimintaympäristöön kuuluvien ohjelmistojen kanssa. Yhteentoimivuusvaatimuksilla otetaan kantaa muun muassa syötteisiin, tulosteisiin sekä käytettäviin protokolliin ja standardeihin. Esimerkki yhteentoimivuuteen liittyvästä vaatimuksesta voisi olla vaikkapa: *Insuliinipumpun suorittamien verensokerimittausten tulokset on voitava avata ja käsitellä Microsoft Excel -muodossa.*

Kuudes ja viimeinen laatuvaatimusten alaluokka on tarkkuus (engl. accuracy) [Lam09, 25]. Tarkkuusvaatimukset määrittelevät sitä, miten tarkasti tulevan järjestelmän ohjelmiston tuottamien tai käsittelemien tietojen tulee vastata todellisuutta. Insuliinipumpun toimintajärjestelmän vaatimusmäärittelyssä tällainen tarkkuusvaatimus voisi olla: *Insuliinipumpun suorittaman verensokerimittauksen tulos saa vaihdella vastaavasta laboratorio-olosuhteissa suoritettuna mittauksen tuloksesta maksimissaan 5 %.*

Toinen ei-toiminnallisten vaatimusten pääluokka on mukautuvuus (engl. compliance) [Lam09, 27]. Mukautuvuusvaatimuksilla määritellään sitä, miten tulevassa järjestelmässä tulee huomioida esimerkiksi kansalliset tai kansainväliset lait ja säädökset, sosiaaliset normit, kulttuurilliset tai poliittiset rajoitukset, standardit ja muut vastaavat tekijät. Insuliinipumpun vaatimusmäärittelyssä mukautuvuusvaatimuksena voisi olla: *Insuliinipumpun toimintajärjestelmässä on voitava määritellä laitteen käyttömaa, ja järjestelmän on tarjottava automaattisesti kyseisessä maassa käytössä olevia mittayksiköitä ja maan terveysviranomaisen määrittelemiä viitearvoja.*

Kolmas pääluokka on arkkitehtuuri (engl. architectural constraint) [Lam09, 27]. Arkkitehtuurivaatimuksilla määritellään, miten tuleva järjestelmä soveltuu ympäristöönsä. Ne jakautuvat kuvan 5 mukaisesti kahteen alaluokkaan: asennettavuus- ja hajautusvaatimuksiin. Asennettavuusvaatimukset (engl. installation) kuvaavat sitä, miten tulevan



järjestelmän käyttöympäristö ja -alusta on huomioitava kehityksessä, jotta järjestelmä toimisi siellä mahdollisimman hyvin. Insuliinipumppuun liittyvä asennettavuusvaatimus voisi olla: *Insuliinipumpun mobiilisovelluksessa tulee olla tuki Android-, iOS, WP7- ja WP8-käyttöjärjestelmille.* Toinen arkkitehtuurivaatimusten alaluokka on hajautus (engl. distribution). Hajautusvaatimukset määrittelevät sitä, kuinka tulevan järjestelmän ohjelmistoissa, laitteissa, muissa komponenteissa sekä käsiteltävissä tiedoissa on huomioitava maantieteellinen hajautus ja sijoittelu. Insuliinipumppuesimerkissämme tällainen vaatimus voisi olla vaikkapa: *Insuliinipumpun suorittamien verensokerin mittaustuloksien tulee tallentua pilvipalveluun, ja niitä tulee voida tarkastella insuliinipumpun näytöltä sekä mobiilisovelluksen kautta tabletilla tai älypuhelimella käyttäjän sijainnista riippumatta.*

Neljäs ja viimeinen ei-toiminnallisten vaatimusten pääluokka on kehitystyön vaatimukset (engl. development constraint) [Lam09, 27-28]. Näiden vaatimusten tarkoituksena ei ole ottaa kantaa siihen, miten tuleva järjestelmä täyttää sille määritellyt toiminnalliset vaatimukset. Sen sijaan niillä asetetaan määriytyksiä, ohjeistuksia ja rajoitteita tulevan järjestelmän kehitystyölle. Kuten kuvasta 5 selviää, kehitystyön vaatimukset jaetaan neljään alaluokkaan: kustannuksiin, tavoiteaikaan, monimuotoisuuteen ja ylläpidettävyyteen. Kustannusvaatimuksilla (engl. cost) asetetaan mahdolliset kustannusraamit, joita kehitysprojektissa tulee noudattaa. Insuliinipumpun toimintajärjestelmän kehitystyössä voitaisiin asettaa esimerkiksi seuraavanlainen kustannusrajoite: *Insuliinipumpun toimintajärjestelmän kehitystyön kokonaiskustannukset eivät saa ylittää 500 000 euroa.* Tavoiteaikavaatimuksilla (engl. deadline) puolestaan määritellään, milloin järjestelmän on oltava valmis [Lam09, 27-28]. Insuliinipumppu-esimerkissä tällainen vaatimus voisi olla: *Insuliinipumpun toimintajärjestelmän on oltava asennettavissa pumppuihin ensi vuoden (2017) huhtikuuhun mennessä.*

Kehitystyölle asetetuilla monimuotoisuusvaatimuksilla (engl. variability) määritellään, missä määrin järjestelmä tulee tarjoamaan erityisominaisuuksia ja -toimintoja eri käyttäjäryhmiä varten [Lam09, 27-28]. Esimerkissämme monimuotoisuuteen liittyvä vaatimus olisi: *Insuliinipumpun toimintajärjestelmän on täytettävä potilaan tarpeiden ohella myös terveydenhuoltohenkilöstön tarpeet.* Tällöin toimintajärjestelmään voitaisiin toteuttaa terveydenhuoltohenkilöstön kirjautumismahdollisuus, jotta he pääsevät tarkastelemaan järjestelmään tallennettuja tietoja. Ylläpidettävyyksivaatimuksilla (engl. main-

tainability) puolestaan ohjeistetaan järjestelmän ylläpitoon ja jatkokehitykseen liittyviä seikkoja. Usein tähän luokkaan lasketaan myös uudelleenkäytettävyyteen sekä siirrettävyyteen liittyvät vaatimukset. Ylläpidettävyyysvaatimus voisi olla esimerkiksi: *Insuliinipumpun toimintajärjestelmän ohjelmiston koodaustyössä tulee kommentoida ja selittää koodia toimintalogiikan ymmärtämiseksi.*

Edellä esiteltiin toiminnallisten ja ei-toiminnallisten vaatimusten luokittelu van Lamsweerden mukaan. Hän kuitenkin korostaa, ettei luokittelu ole aina täysin yksiselitteistä, sillä joskus jokin vaatimus voi olla luonteeltaan toisaalta toiminnallinen vaatimus ja toisaalta ei-toiminnallinen vaatimus [Lam09, 28]. Esimerkkinä tällaisesta tilanteesta on laatuvaatimukseen lukeutuva luotettavuusvaatimus: *Jos insuliinipumpun verensokerimitaus epäonnistuu, täytyy käyttäjän siitä huolimatta voida annostella insuliinia manuaalisesti.* Vaatimusta voidaan ajatella toiminnallisena vaatimuksena, eli käyttäjän on voitava annostella insuliinia myös manuaalisesti laitteen toiminnasta tai toimimattomuudesta huolimatta. Toisaalta kyse on myös luotettavuudesta, eli laitteen toimintavirhe ei voi estää käyttäjää annostelemasta insuliinia manuaalisesti. Van Lamsweerde korostaakin, ettei vaatimusten oikea kategorisointi ole niinkään tärkeää, vaan olennaista on pyrkiä tunnistamaan vaatimuksia mahdollisimman monesta näkökulmasta.

## 2.4 Vaatimusmäärittelyn laatu ja onnistuminen

Kuten jo aiemmin todettiin, vaatimusmäärittelyn epäonnistuminen ja epäselvät vaatimukset ovat yksi yleisimmistä syistä tietojärjestelmäprojektien epäonnistumiseen [The95, 4-5 ja FGZ15, 26]. Tätä selittää omalta osaltaan kuvassa 4 konkretisoitunut tosiasia, että vaatimusdokumenttia hyödynnetään kehitysprojektissa hyvin monella tavalla. Jos vaatimukset ovat puutteellisia tai jopa virheellisiä, sillä on laajoja vaikutuksia niin itse kehitysprojektiin kuin valmiisiin lopputuotoksiinkin. Kaikissa tietojärjestelmi- en kehitysprojekteissa tulisi siis panostaa vaatimusmäärittelyn laatuun.

Millaista laadukas vaatimusmäärittely sitten on? Van Lamsweerde [Lam09, 35-36] esittelee listan laatutavoitteista, joihin vaatimusmäärittelydokumentaatioissa tulee pyrkiä. Ensimmäisenä hän mainitsee *täydellisyyden* (engl. completeness). Täydellisyydellä tarkoitetaan tässä sitä, että vaatimusmäärittelyprosessissa on pystytty tunnistamaan kaikki tulevaan järjestelmään kohdistuvat tarpeet ja asettamaan niiden pohjalta tavoitteet tulevalle järjestelmälle. Myös kaikki mahdolliset virhe- ja poikkeustilanteet on pystytty

ennakoimaan ja asettamaan tarvittaessa niiden käsittelyyn liittyviä vaatimuksia. Lisäksi kaikki mahdolliset tulevan järjestelmän syötteet ja niihin vastauksena olevat tulosteet on tunnistettu.

Toinen dokumentaatioon liittyvä laatutavoite van Lamsweerden [Lam09, 35] mukaan on *yhdenmukaisuus* (engl. consistency). Yhdenmukaisuudella hän tarkoittaa sitä, että kaikkien tunnistettujen vaatimusten, oletusten ja reunaehtojen tulee muodostaa järkevä kokonaisuus eli olla keskenään yhteensopivia. Kolmas laatutavoite on *sopivuus* (engl. adequacy). Sen mukaisesti dokumentoitujen vaatimusten tulee kuvata niitä todellisia tarpeita, joita sidosryhmillä on suhteessa tulevaan järjestelmään. Sopivuus edellyttää myös sitä, että ohjelmistovaatimukset ovat linjassa järjestelmävaatimusten kanssa ja että reunaehdot kuvaavat täsmällisesti ongelmakentän lainalaisuudet. Lisäksi toimintaympäristöön liittyvien oletusten tulee olla realistisia.

Neljäs laatutavoite vaatimusmäärittelydokumenteille on *yksiselitteisyys* (engl. unambiguity) [Lam09, 35]. Vaatimukset, oletukset ja reunaehdot tulee kuvata niin, ettei niitä voi tulkita väärin. Käytetyt termit tulee määritellä esimerkiksi sanastossa, ja termien käytön tulee olla johdonmukaista. Viides laatutavoite on *mitattavuus* (engl. measurability). Mitattavuudella van Lamsweerde tarkoittaa sitä, että vaatimukset tulee määritellä ja kirjata sellaisella tarkkuudella, että niiden keskinäinen vertailu ja arviointi on mahdollista. Sen myötä on myös mahdollista arvioida toteutuuko jokin tietty vaatimus vai ei.

Kuudentena laatutavoitteena van Lamsweerde [Lam09, 35] mainitsee *asianmukaisuuden*. Sillä hän tarkoittaa sitä, että vaatimusten ja oletusten tulee liittyä tulevan järjestelmän tavoitteisiin ja ongelmakenttään, ei toteutustapaan tai toteutuksen yksityiskohtiin. Seitsemäs laatutavoite on *kelpoisuus* (engl. feasibility). Kelpoisuus kuvastaa vaatimusten realistisuutta suhteessa annettuun budjettiin, aikatauluun ja käytettävissä olevaan teknologiaan. Kahdeksantena laatutavoitteena on *ymmärrettävyys* (engl. comprehensibility). Ymmärrettävyyden mukaan kaikkien vaatimusten, oletusten ja reunaehtojen tulee olla kuvattu niin selkeästi ja tarkasti, että kaikki niiden parissa työskentelevät ymmärtävät ne oikein.

Yhdeksäs laatutavoite on *hyvä rakenne* (engl. good structuring) [Lam09, 36]. Vaatimusmäärittelydokumentaatioissa hyvä rakenne tarkoittaa sitä, että rakenne on looginen ja yhtenäinen. Elementtien keskinäisten yhteyksien kuten riippuvuuksien ja syy-seuraus

-suhteiden tulee olla kuvattu selkeästi. Lisäksi kaikki termit tulee määritellä ennen niiden käyttöä. Kymmenentenä laatutavoitteena van Lamsweerde mainitsee *muunneltavuuden* (engl. modifiability). Kun vaatimusmäärittelydokumentti on muunneltava, sitä on helppo päivittää, laajentaa tai supistaa tarpeen vaatiessa. Yhdestoista ja viimeinen laatuvaatimus on *jäljitettävyys* (engl. traceability). Jäljitettävyydellä tarkoitetaan sitä, että jokaisen vaatimusdokumenttiin kirjatun vaatimuksen, oletuksen ja reunaehdon esittäjätaho ja perustelut täytyy selvittää dokumentista. Lisäksi edellä mainittujen elementtien keskinäiset suhteet ja riippuvuudet sekä yhteydet dokumentin ulkopuolisiin elementteihin, kuten testitapauksiin, lähdekoodiin tai arkkitehtuurikuvauksiin tulee dokumentoida selvästi. Käytännössä jäljitettävyys mahdollistaa sen, että vaatimusmäärittelyprosessin aikana voidaan helposti arvioida, mihin kaikkialle jonkin vaatimuksen muuttaminen tai poistaminen vaikuttaa.

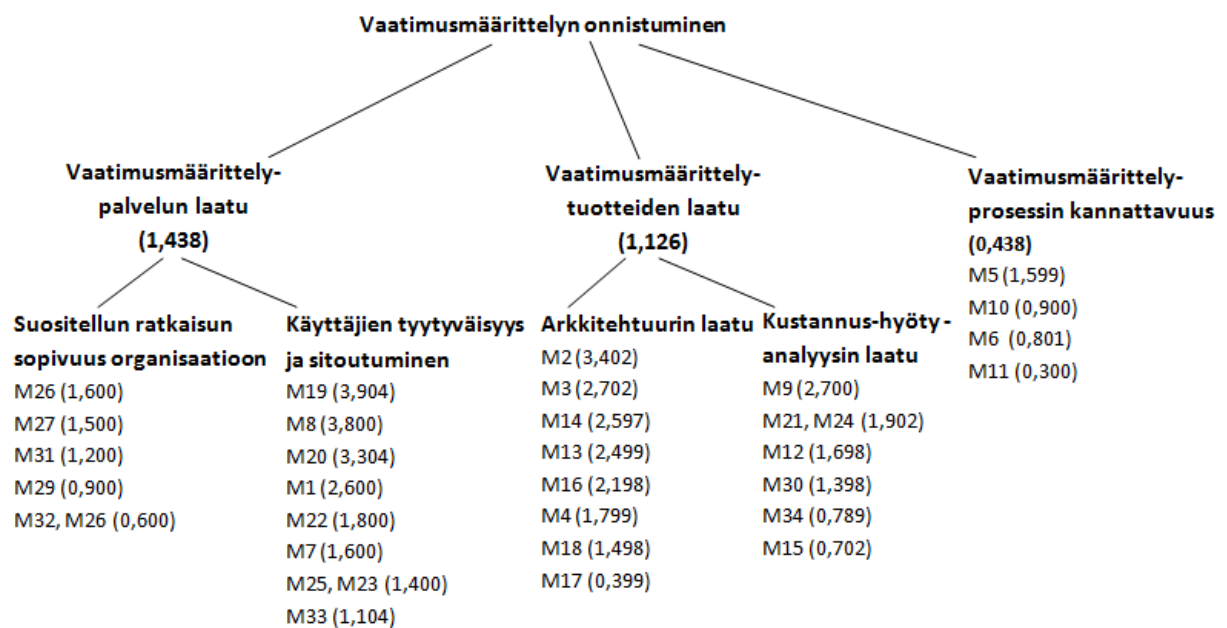
Van Lamsweerde korostaa, etteivät laatutavoitteet ole absoluuttisia tai yhtä kriittisiä kaikissa kehitysprojekteissa [Lam09, 36]. Laatutavoitteet tulisikin määritellä aina tapauskohtaisesti suhteessa kehitettävän järjestelmän tarpeisiin ja erityispiirteisiin. Esimerkiksi turvallisuuskriittisten järjestelmien vaatimusmäärittelyssä on hyvin olennaista, että kaikki vaatimukset on tunnistettu kattavasti eli vaatimusmäärittelydokumentti täyttää täydellisyysvaatimuksen. Olisi nimittäin hyvin ongelmallista, jos insuliinipumpun vaatimuksista olisi kokonaan unohdettu jokin olennainen vaatimus tai jätetty huomiotta jonkin kriittisen virhetilanteen käsittely. Käyttäjä joutuu elämään puutteellisen tuotteen kanssa eikä häntä paljoa lohduta, että ominaisuus luvataan päivittää tulevaan versioon. Kyseessä kun on hänen terveytensä ja elämänsä. Vastaavasti jonkin pelisovelluksen suhteen täydellisyysvaatimus ei ole niin oleellinen. Jos jokin tärkeä vaatimus unohtuu, se voidaan suhteellisen helposti päivittää seuraavaan versioon. Suurin kuviteltava tappio on rahallinen, ja pelaajat ovat todennäköisesti muutenkin tottuneet siihen, että peleistä julkaistaan uusia, toimivampia versioita ajan mittaan.

Joskus kehitettävään järjestelmään kohdistuvat tarpeet on määritelty implisiittisesti tai epäselvästi [Lam09, 36]. Tällöin laatutavoitteiden asettaminen tai toteutumisen mittaaminen ei ole aivan yksiselitteistä. Esimerkiksi jos sidosryhmäanalyysissä ei ole tunnistettu kaikkia olennaisia sidosryhmiä, miten todenmukainen on arvio yhdenmukaisuustavoitteen toteutumisesta? Aluksi toki voi vaikuttaa siltä, että vaatimuksissa on päästy riittävään yhdenmukaisuuteen eli kaikki vaatimukset ja oletukset muodostavat järkevän

kokonaisuuden. Jos sitten myöhemmin tunnistetaan uusia sidosryhmiä ja saadaan niiltä uusia – mahdollisesti ristiriitaisia – vaatimuksia, tilanne voi muuttua radikaalistikin. Vaatimuksia ja niiden dokumentointiin kohdistuvia laatutavoitteita tuleekin arvioida ja päivittää projektin aikana säännöllisesti.

Vaatimusmäärittelydokumentaation laadun ohella voidaan arvioida myös itse vaatimusmäärittelyprosessin laatua ja onnistumista. El Emam ja Madhavji [EIM95, 204] ovat kehittäneet menetelmän, jonka avulla he ovat mitanneet vaatimusmäärittelyprosessien onnistumista. Heidän mukaansa vaatimusmäärittelyn onnistumisella tarkoitetaan sitä, miten hyvin vaatimusmäärittelyn tuotokset palvelevat ja pohjustavat tietojärjestelmien kehitysprosessin myöhempien vaiheiden ja järjestelmän koko elinkaaren aktiviteettien tarpeita.

El Emam ja Madhavji [EIM95, 208-209] löysivät tutkimuksiensa pohjalta 34 osatekijää, jotka vaikuttavat vaatimusmäärittelyn onnistumiseen. Kaikki osatekijät on lueteltu liitteessä 1, tässä esitellään vain kaikkein merkittävimmät tekijät. Tutkijat havaitsivat, että onnistumistekijät voidaan jakaa kolmeen pääkategoriaan: *vaatimusmäärittelypalvelun laatuun* (engl. quality of RE service), *vaatimusmäärittelyn tuotteiden laatuun* (engl. quality of RE products) sekä *vaatimusmäärittelyprosessin kannattavuuteen* (engl. cost effectiveness of RE process). Vaatimusmäärittelypalvelun laatu kattaa sekä käyttäjien tyytyväisyyden ja sitoutumisen että suositellun ratkaisun (sillä hetkellä arviointivaiheessa tulevaisuuden ratkaisuksi valittu ratkaisu) sopivuuden kyseiseen organisaatioon. Tuotteiden laadun kannalta tarkastelu kohdistuu vaatimusmäärittelytyön aikana tuotettuihin dokumentteihin, erityisesti arkkitehtuuridokumentaation arviointiin sekä kustannus-hyöty -analyysiin. Vaatimusmäärittelyprosessin kannattavuutta arvioidaan sen pohjalta, käytettiinkö vaatimusmäärittelyyn resursseja järkevästi. Kategoriat (lihavoituna) ja niihin sisältyvät onnistumisen osatekijät (M1-M34) priorisoituna tärkeysjärjestykseen esitellään kuvassa 6.



Kuva 6. Vaatusmäärittelyprosessin onnistumistekijät El Emamin ja Madhavjin mukaan [EIM95, 209 mukailleen].

Kuvassa 6 kategorioiden ja osatekijöiden kohdalla esitetyt luvut kuvastavat sitä, miten paljon kukin kategoria tai osatekijä vaikuttaa osaltaan ylemmän tason kategorian onnistumiseen [EIM95, 209]. Suluissa on painoarvot, joissa suurempi luku kuvastaa suurempaa vaikutusta. Esimerkiksi kolmesta pääkategoriasta eniten vaikutusta vaatusmäärittelyn onnistumiseen kokonaisuutena on *vaatusmäärittelypalvelun laadulla* (painoarvo 1,438). Vähiten vaikutusta puolestaan oli *vaatusmäärittelyprosessin kannattavuudella* (painoarvo 0,438). Näin ollen vaatusmäärittelyssä kannattaa panostaa erityisesti niihin osatekijöihin, jotka liittyvät *vaatusmäärittelypalvelun laatuun* (eli alakategorioihin *suositellun ratkaisun sopivuus organisaatioon* sekä *käyttäjien tyytyväisyys ja sitoutuminen*).

Vastaavasti alakategorioiden onnistumiseen vaikuttavat tekijät on listattu niiden vaikutuksen mukaan [EIM95, 209]. Esimerkiksi alakategoriaan *käyttäjien tyytyväisyys ja sitoutuminen* oli suurin vaikutus osatekijällä M19 (painoarvo 3,904). Osatekijä M19 kuvastaa sitä, miten hyvin käyttäjät ymmärtävät, mitä tuleva järjestelmä tekee ja mitä se ei tee. Käytännössä siis karsimalla vaatusmäärittelytyön aikana väärinymmärryksiä ja turhia odotuksia voidaan tukea vaatusmäärittelyn onnistumista.

Kunkin alakategorian kolme tärkeintä onnistumistekijää on esitelty tarkemmin taulukossa 1. Selkeyden vuoksi onnistumistekijät on esitetty lähdemateriaalista poiketen kysymysmuodossa.

<i>Alakategoria</i>	<i>1. tärkein onnistumistekijä</i>	<i>2. tärkein onnistumistekijä</i>	<i>3. tärkein onnistumistekijä</i>
<i>Suosittelun ratkaisun sopivuus organisaatioon</i>	M26 = Onko organisaatiolla kyky tehdä tarvittavia muutoksia suositellun ratkaisun käyttöönottamiseksi?	M27 = Sopiiko suositeltu ratkaisu yhtiön organisaation strategian kanssa?	M31 = Tukeeko ylin johto tarvittavien muutosten tekemistä suositellun ratkaisun käyttöönottamiseksi?
<i>Käyttäjien tyytyväisyys ja sitoutuminen</i>	M19 = Ovatko käyttäjät ymmärtäneet, mitä tuleva järjestelmä tekee ja mitä se ei tee?	M8 = Ovatko käyttäjät halukkaita puolustamaan suositeltua ratkaisua johdolle?	M20 = Vallitseeko käyttäjien kesken yhteisymmärrys suositellusta ratkaisusta?
<i>Arkkitehtuurin laatu</i>	M2 = Onko mallien ja tietojärjestelmän tavoitteiden väliset yhteydet kuvattu selkeästi sekä prosessien että tietojen kannalta?	M3 = Kuvastuuko liiketoimintaprosessi selkeästi arkkitehtuurissa?	M14 = Onko nykyinen tietojärjestelmä analysoitu kattavasti?
<i>Kustannus-hyöty-analyysin laatu</i>	M9 = Onko kustannus-hyöty -analyysi kattava?	M21 = Onko ylin johto vakuuttunut siitä, että tavoitellut hyödyt tulevat toteutumaan?	M24 = Onko käytettävissä oleva rahoitusmalli yhteneväinen suositeltuun ratkaisuun tarvittavan rahoitusmallin kanssa?
<i>Vaatusmäärittely-prosessin kannattavuus</i>	M5 = Onko vaatimusmäärittelytyön tuotos-panos -suhde yhteneväinen organisaation omien tai muiden vastaavalaisten organisaatioiden projekteihin?	M10 = Mikä osuus koko projektin kehittämiskustannuksista varataan vaatimusmäärittelytehtäviin?	M6 = Paljonko vaatimusmäärittelydokumentaatioon tehdään muutoksia?

Taulukko 1. Kolme tärkeintä vaatimusmäärittelyn onnistumiseen vaikuttavaa osatekijää alakategorioittain El Emamin ja Madhavjin mukaan [ElM95, 208-209 mukailen].

Esimerkiksi taulukon 1 mukaisesti *suositellun ratkaisun sopivuus organisaatioon* -kategoriassa eniten on merkitystä onnistumistekijöillä M26, M27 ja M31. Niiden myötä vaatimusmäärittelyn onnistumiseen vaikuttaa se, että organisaatiolla on kykyä ja halua

tehdä tarvittavat muutokset suositellun ratkaisun käyttöönottamiseksi, että ratkaisu on yhteensopiva organisaation strategian kanssa ja että suositellun ratkaisun käyttöönottamista edellyttävillä muutoksilla on ylimmän johdon tuki. Vastaavasti *kategoriaan käyttäjien tyytyväisyys ja sitoutuminen* eniten vaikuttavat tekijät M19, M8 ja M20. Onnistumisessa on merkitystä sillä, että käyttäjät ovat ymmärtäneet, mitä suositeltu ratkaisu tekee ja mitä se ei tee, että käyttäjät ovat valmiita puolustamaan suositeltua ratkaisua johdolle ja että he ovat yksimielisiä suositellusta ratkaisusta.

Kuten taulukosta 1 selviää, *arkkitehtuurin laatu* -kategoriaan on eniten vaikutusta onnistumistekijöillä M2, M3 ja M14. Niiden mukaisesti vaatimusmäärittely voi onnistua, jos tietojärjestelmän tavoitteet ja mallit ovat yhtenäisiä, jos arkkitehtuuri kuvastaa liiketoimintaprosessia ja jos nykyistä järjestelmää on analysoitu riittävän tarkalla tasolla. *Kustannus-hyöty -analyysin laatu* -kategorian olennaisimmat tekijät ovat M9, M21 ja M24. Niiden kannalta vaatimusmäärittely voi onnistua, kun kustannus-hyöty -analyysi on kattava, kun ylin johto uskoo hyötyjen realisoituvan ja kun rahoitussuunnitelma on realistinen ja sopiva. Lopuksi *vaatimusmäärittelyprosessin kannattavuus* -kategoriassa eniten vaikutusta on onnistumistekijöillä M5, M10 ja M6. Jos kyseisen kehitysprojektin oletettu panos-tuotos -suhde on linjassa organisaation muiden vastaavien projektien kanssa, jos riittävän suuri osa kehityskustannuksista on allokoitu vaatimusmäärittelytyöhön ja jos vaatimusdokumentaation muutostarpeet ovat maltilliset, voidaan vaatimusmäärittelyssä onnistua.

El Emam ja Madhavji eivät ole määritelleet onnistumistekijöitä absoluuttisesti. Niitä ei siten voi sisällyttää sellaisenaan uuteen kehitysprojektiin ja olettaa automaattisesti sen vaatimusmäärittelyn onnistuvan. Tarkastellaan esimerkiksi onnistumistekijää M10, jonka mukaan vaatimusmäärittelytehtävien kustannusten osuus koko projektin kustannuksista vaikuttaa vaatimusmäärittelyn onnistumiseen. Kyseiselle onnistumistekijälle ei ole asetettu mitään tavoiteprosenttia tai osuutta, miten paljon resursseja tai aikaa vaatimusmäärittelytyöhön pitäisi varata projektin kokonaisbudjetista. Olennaisempaa onkin tiedostaa, että varaamalla kokonaiskustannuksista ja resursseista riittävän suuri osa vaatimusmäärittelytyöhön mahdollistetaan vaatimusmäärittelyn onnistuminen. Vastaavasti kiinnittämällä muutenkin erityistä huomiota taulukossa 1 esiteltyihin merkittävimpiin osatekijöihin voidaan parantaa vaatimusmäärittelytyön onnistumismahdollisuuksia



huomattavasti. Sen sijaan pienemmän painoarvion omaaviin onnistumistekijöihin ei välttämättä tarvitse panostaa erityisesti.

Kuvassa 6 esiteltyjä onnistumistekijöitä ja niiden painoarvoja voidaan kuitenkin hyödyntää kehitysprojektien ja niiden vaatimusmäärittelyvaiheen suunnittelussa. Onnistumistekijöitä kannattaa analysoida suhteessa organisaatioon, lopputuotteeseen ja kehitysprojektiin, ja valita juuri kyseisessä tapauksessa keskeiset tekijät. Niiden pohjalta rakennetaan vaatimusmäärittelyn strategia; mitkä onnistumistekijät ovat avainasemassa eli mihin on syytä panostaa ja miten ja mitkä tekijät voidaan tietoisesti jättää vähemmälle huomiolle. Tämä mahdollistaa resurssien järkevän kohdistamisen ja tukee vaatimusmäärittelyn onnistumista.

Hofmann ja Lehner [HoL01, 61-62] vievät tarkastelun konkreettisemmalle tasolle kuvaamalla vaatimusmäärittelyn parhaita käytäntöjä. He ovat tutkimuksissaan tunnistaneet kolme ulottuvuutta, jotka vaikuttavat vaatimusmäärittelyn onnistumiseen. Ensimmäinen ulottuvuus on *tieto*. Tiedolla tarkoitetaan tässä sekä kehitystiimin että sidosryhmien alatietyä ja kokemusta. Toinen ulottuvuus on *resurssit* eli kuinka paljon resursseja ja aikaa vaatimusmäärittelytyöhön varataan. Kolmas ulottuvuus on *prosessi* eli miten vaatimusmäärittelytyö projektissa organisoidaan. Näiden kolmen ulottuvuuden osalta Hofmann ja Lehner esittelevät vaatimusmäärittelyn parhaat käytännöt, jotka parantavat vaatimusmäärittelyn laatua ja onnistumismahdollisuuksia.

Hofmann ja Lehner [HoL01, 65] esittelevät kolme tietoon liittyvää parasta käytäntöä. Ensimmäiseksi; asiakkaat ja käyttäjät tulee osallistaa vaatimusmäärittelyprosessiin alusta loppuun asti. Tällä voidaan auttaa varmistamaan, että projektissa ymmärretään tulevaan järjestelmään kohdistuvat todelliset tarpeet. Toiseksi; vaatimusmäärittelyssä tulee myös tunnistaa ja hyödyntää kaikki mahdolliset vaatimuslähteet. Näin tulevaa järjestelmää voidaan tarkastella mahdollisimman monesta näkökulmasta ja parannetaan vaatimusten kattavuutta. Kolmanneksi; vaatimusmäärittelytehtäviin tulisi kiinnittää osaava ja kokenut projektipäällikkö ja tiimi, jolloin suorituksen taso on ennakoitavissa.

Resursseihin liittyen Hofmann ja Lehner [HoL01, 65] ovat tunnistaneet kolme parasta käytäntöä. Ensimmäisenä he ehdottavat, että vaatimusmäärittelytehtäviin tulisi kohdistaa noin 15-30 % projektin kokonaispanostuksesta. Ideaalitapauksessa panostus olisi 28 % jakautuen vaatimusmäärittelytehtäville seuraavasti: 11 % vaatimusten kartutukseen,

10 % mallinnukseen ja 7 % validointiin sekä verifiointiin. Toinen resursseihin liittyvä paras käytäntö on mallipohjien ja esimerkkien hyödyntäminen spesifikaatioiden tuottamisessa. Tämä nopeuttaa työskentelyä ja parantaa spesifikaatioiden laatua määrämuitoisten, helposti luettavien ja ymmärrettävien lopputuotosten myötä. Kolmas paras käytäntö resurssinäkökulmasta on hyvien suhteiden säilyttäminen sidosryhmien kanssa. Kun vuorovaikutus on säännöllistä ja luonteeltaan positiivista, on palautteen antaminen ja saaminen helpompaa. Avoimella ilmapiirillä kannustetaan eri näkökulmien esille tuomiseen ja voidaan välttää väärinymmärryksiä. Näin asiakkaiden tarpeet saadaan esille selkeämmin, ja ne voidaan myös täyttää paremmin.

Prosesseihin liittyviä parhaita käytäntöjä on tunnistettu neljä [HoL01, 65-66]. Ensimmäisenä tutkijat mainitsevat vaatimusten priorisoinnin. Priorisoinnin avulla saadaan selville, mitkä vaatimukset ovat asiakkaiden kannalta tärkeimpiä, jolloin resurssit voidaan kohdentaa niiden työstämiseen. Toisena prosesseihin liittyvänä parhaana käytäntönä Hofmann ja Lehner suosittelevat täydentävien mallien ja prototyyppien käyttöä. Niiden avulla voidaan esitellä asiakkaille tulevan järjestelmän toimintaa ja ominaisuuksia konkreettisella tasolla. Tämä helpottaa määrittelyihin liittyvien epäjohtonmukaisuuksien ja monitulkintaisuuksien tunnistamista ja eliminointia. Kolmas prosessinäkökulman paras käytäntö on jäljitettävyyssmatriisin laatiminen ja ylläpito. Kun vaatimukset ja reunaehdot on linkitetty spesifikaatioihin ja niiden toteutukseen, voidaan helposti todentaa, minne kaikkialle mahdollinen muutos vaatimuksissa vaikuttaa. Jäljitettävyys auttaa myös osoittamaan, missä määrin lopputuotteet täyttävät esitetyt vaatimukset. Neljäs ja viimeinen paras käytäntö prosessien osalta on katselmointien, läpikäyntien ja skenaarioiden käyttö vaatimusten validoinnissa ja verifiointissa. Näin spesifikaatioista kehittyy tarkempia ja asiakastyytyväisyys paranee.

Hofmannin ja Lehnerin [HoL01, 65] esittelemien vaatimusmäärittelyn parhaiden käytäntöjen avulla kehitysprojekteissa voidaan siis pyrkiä parantamaan vaatimusmäärittelyn onnistumismahdollisuuksia. Joitakin käytännöistä on suhteellisen helppoa ja edullista soveltaa; esimerkiksi mallipohjien laatiminen ja käyttö ei tuo kovin suuria lisäkustannuksia organisaatiolle ja projektille. Toiset käytännöistä puolestaan vaativat enemmän suunnittelua ja aiheuttavat kuluja. Miten organisaatiossa voidaan esimerkiksi huolehtia siitä, että kaikkiin käynnissä oleviin kehitysprojekteihin riittäisi osaavaa ja kokenutta henkilöstöä? Organisaation tulisi aloittaa henkilöstön koulutukset ja tarvittavat rekry-

toinnit hyvissä ajoin, ja siltikin toimenpiteet kantavat hedelmää vasta useamman vuoden päästä, kun henkilöstölle karttuu kokemusta työtehtävien myötä. Yhtenä vaihtoehtona Hofmann ja Lehner tarjoavat tähän "osa-aikaisten konkareiden käyttöä" [HoL01, 64]. Organisaation syväosaajat voisivat osallistua useaan kehitysprojektiin pienemmällä työpanoksella tarjoten omaa osaamistaan projektien käyttöön katselmoimalla tai antamalla tarvittaessa konsultaatiota. Tällöin projektin täysipäiväiset työntekijät voivat olla vähemmän kokemusta omaavia, mikä helpottaa resurssiongelmaa.

Hofmannin ja Lehnerin parhaat käytännöt samoin kuin edellä esitellyt El Emamin ja Madhavjin onnistumistekijät edellyttävät siis strategista etukäteissuunnittelua. Organisaatiolla tulisi olla näkemys, miten ja missä rajoissa mihinkin tekijään pitää ja voi panostaa. Lisäksi kehitysprojektien esiselvitysvaiheessa tulee vielä tarkastaa, mitkä onnistumistekijät tai parhaat käytännöt ovat niitä, joihin juuri kyseisessä projektissa kannattaa keskittyä. Onnistumistekijöihin ja parhaisiin käytäntöihin satsaaminen ei ole varma tae vaatimusmäärittelyn ja koko kehitysprojektin onnistumiselle, mutta lienee selvää, että ne mahdollistavat onnistuneen lopputuloksen varmemmin kuin pelkkään onneen luottaminen.

### **3 Sähköinen lääkemääräys sähköisen terveydenhuollon esimerkkitapauksena**

Tässä luvussa kerrotaan taustatietoa sähköisestä terveydenhuollosta ja siihen liittyvistä erityispiirteistä. Alan esimerkkitapauksena tutustutaan sähköiseen lääkemääräykseen sekä siihen liittyviin hyötyihin ja ongelmiin. Lisäksi luodaan katsaus erääseen yleiseen tietojärjestelmäratkaisuun, jonka mukaisesti sähköinen lääkemääräysjärjestelmä voidaan toteuttaa. Luvussa kuvataan myös Suomessa vuonna 2010 käyttöön otetun sähköisen lääkemääräyksen tietojärjestelmäratkaisu sekä sen käyttäjilleen tarjoamat toiminnallisuudet.

#### ***3.1 Sähköisen terveydenhuollon historia ja erityispiirteet***

Termillä sähköinen terveydenhuolto (engl. digital health, e-Health) tarkoitetaan laajaa joukkoa erilaisia diagnostisia ja interventionaalisia teknologioita ja sovelluksia [Thü15, 13]. Sähköisen terveydenhuollon tietojärjestelmiä voidaan kategorisoida luvussa 2.1

esitellyn Pohjosen tietojärjestelmäluokittelun [Poh05, 338-339] mukaisesti. *Toimisto-automaatiojärjestelmiä* ovat sähköiset potilaskertomus- ja potilastietojärjestelmät (engl. electronic health record eli EHR tai electronic medical record EMR), kun niiden avulla luodaan, käsitellään ja säilytetään potilaisiin liittyviä asiakirjoja ja muita tallenteita. *Tapahtumankäsittelyjärjestelmiksi* voidaan luokitella potilasrekisterit ja ajanvarausjärjestelmät. *Reaaliaikaisia järjestelmiä* edustaa insuliinipumppu, joka huolehtii käyttäjänsä insuliinitasapainosta ruiskuttamalla automaattisesti insuliinia ihon alle tarpeen mukaan ympäri vuorokauden. Sähköinen lääkemääräys voidaan kategorisoida *päätöksentekijärjestelmäksi*, sillä se avustaa lääkärinä ja apteekkaria valitsemaan oikean lääkkeen tai annostuksen huomioiden esimerkiksi potilaan järjestelmään tallennetut allergiat ja yleiset annosteluohjeet. Lisäksi terveydenhuoltohenkilöstön toiminnan tueksi on olemassa kliinisen lääketieteen eri osa-alueille suunnattuja *asiantuntijajärjestelmiä*. Tällaisia järjestelmiä ovat esimerkiksi eri alueiden hoito-ohjeita ja -suosituksia tarjoavat järjestelmät tai diagnoosien tekemistä tukevat järjestelmät.

Thümmelerin [Thü15, 13] mukaan termiä sähköinen terveydenhuolto käytetään usein hyvin laajassa merkityksessä. Sillä voidaan viitata perinteisen terveydenhuollon lisäksi esimerkiksi erilaisiin hyvinvointiin liittyviin tai vaikkapa ikääntyvien kansalaisten asumismukavuutta parantaviin sovelluksiin. Tyypillisinä esimerkkeinä voidaan mainita kannettavat insuliinipumput, älypuhelimien mobiilisovellukset ja päätöksentekoa tukevat potilastietojärjestelmät. Suomenkielessä sähköisen terveydenhuollon ohella näkee usein käytettävän termejä teleterveydenhuolto, telelääketiede tai sähköiset terveystalvut. Yhteisenä nimittäjänä kaikille edellä mainituille termeille on se, että tavalla tai toisella ne hyödyntävät ICT-alan saavutuksia, kuten tietojenkäsittelyä, digitaalisia viestintäverkkoja tai tekoälyä. Tarkastelunäkökulma hieman vain vaihtelee.

Tässä tutkielmassa keskitytään sähköisen terveydenhuollon osalta nimenomaan terveydenhuoltoon liittyviin ratkaisuihin ja sovelluksiin sekä erityisesti niiden vaatimusmäärittelyyn. Tarkastelun keskiössä ei ole se, lähestytäänkö ilmiötä terveydenhuollon, lääketieteen, teknologian vai kenties asiakkaille tarjottavien palveluiden näkökulmasta. Sen sijaan olennaista on se, että kyseessä on käyttäjien ja asiakkaiden terveys ja siihen liittyvät arkaluonteiset tiedot. Tällöin on ensisijaisen tärkeää, että sähköisen terveydenhuollon sovellusten toiminta on määritelty tarkasti ja kattavasti ja että sovellukset myös toimivat määrittystensä mukaisesti ja luotettavasti.

Thümmeler [Thü15, 13] on selvittänyt sähköisen terveydenhuollon historiaa. Hän tuo esille, että kehitys on ollut varsin nopeaa, mutta ei yksiselitteisen suoraviivaista. Kehitysprosessiin ovat vaikuttaneet omalta osaltaan teknologian kehittymisen tarjoamat mahdollisuudet, mutta niitäkin voimakkaammin sosioekonomiset ja poliittiset tekijät [Säh12, 16-17]. Thümmelerin [Thü15, 13] mukaan sähköisen terveydenhuollon juuret ulottuvat aina 1960-luvulle. Tuolloin ICT-teknologian hyödyntäminen terveydenhuollon alueella tarkoitti lähinnä sairaaloiden henkilöstön palkanmaksun, henkilöstöressurssien sekä varastotilausten hallintaa keskustietokoneiden (engl. mainframe) avulla. Vuosien saatossa 1960- ja 1970-luvuilla terveydenhuollon palvelut laajenivat ja tulivat yhä useamman saataville. Terveydenhuoltopalvelujen tarjoajien oli pystyttävä käsittelemään potilastietoa nopeasti ja välittämään sitä niin potilaiden, terveydenhuollon ammattilaisten kuin vakuutusyhtiöidenkin välillä. Sairaaloissa alettiin siirtää erilaisia operationaalisia prosesseja, kuten kirjanpitoa ja potilaiden vakuutustietojen hallintaa keskustietokoneilla ja reikäkorteilla suoritettaviksi.

Sähköisen terveydenhuollon alueella 1980- ja 1990-luvut olivat voimakkaan kehityksen aikaa [Thü15, 14]. Ajan merkittäviä kehitysaskelaita olivat muun muassa digitaaliset kuvat ja niihin liittyvä DICOM-standardi (Digital Imaging and Communications in Medicine). Standardi mahdollisti esimerkiksi magneettikuvien ja ultraäänikuvien tallentamisen ja käsittelyn tietokoneella. Samaan aikaan yleistynyt asiakas-palvelin -arkkitehtuuri mahdollisti erilaisten tietojärjestelmäratkaisuiden kehittämisen yksittäisiin tarpeisiin. Koska kehitys ei ollut mitenkään organisoitua tai säädeltyä, se johti eristäytymisiin tietojärjestelmäsaarekkeisiin ja pirstoutumiseen, eivätkä järjestelmät olleet millään lailla keskenään yhteentoimivia (engl. interoperable).

Bourguardin, Gallin ja Cousinin mukaan yhteentoimimattomuus tuottaa edelleenkin suuria haasteita sähköisen terveydenhuollon monimutkaisissa tietojärjestelmissä [BLC15, 95-97, 110]. Haasteisiin on yritetty vastata kehittämällä erilaisia standardeja, kuten HL7:n (Health Level 7) V3 -sanomavälitysstandardi. Kun standardeja noudatetaan, on järjestelmillä edes hypoteettiset mahdollisuudet välittämään tietoa ja olla yhteentoimivia keskenään. Mistä sitten voi tietää, mitä standardeja tulevan järjestelmän kehitystyössä tulisi huomioida ja noudattaa? IHE (Integrating the Healthcare Enterprise) on kehittänyt avuksi järjestelmien integraatiota helpottavia profiileja. Niitä noudattamalla voidaan varmistua siitä, että tuleva ratkaisu tulee olemaan laajalti yhteentoimiva vastaa-

vaa profiilia noudattavien tietojärjestelmien kanssa. Esimerkiksi IHE:n XDR-profiili (Cross-Enterprise Document Reliable Interchange) määrittelee viitekehyksen ja noudattavat standardit luotettavalle kliinisten dokumenttien välitykselle organisaatioiden välillä. Kun järjestelmät on toteutettu profiilin mukaisesti, ne pystyvät keskustelemaan keskenään ja välittämään tietoa järjestelmästä toiseen. Alan standardit ja profiilit tuleekin ehdottomasti huomioida jo vaatimusmäärittelyvaiheessa, jotta ne voidaan huomioida kehitystyön alusta alkaen eikä tehdä turhaa työtä.

Matkapuhelimet ja langattomat verkot yleistyivät 1990-luvun lopulla [Thü15, 14]. Ne toivat mukanaan uuden tavan hyödyntää ICT-teknologiaa terveydenhuollon alueella: syntyi käsite mobiiliterveydenhuolto (engl. m-Health). Mobiiliterveydenhuollolla tarkoitetaan ajasta ja paikasta riippumatonta tapaa käyttää sähköisen terveydenhuollon sovelluksia mobiililaitteilla. Käytännössä mobiiliterveydenhuolto on siis osa sähköistä terveydenhuoltoa. 2000-luvulla yleistyneiden älypuhelimien ja tablettitietokoneiden myötä mobiiliterveydenhuolto on lähestulkoon kaikkien saatavilla, ja yhä useammat kansalaiset käyttävät erilaisia omaan terveydentilaan ja hyvinvointiin liittyviä mobiili-sovelluksia.

Mobiiliterveydenhuollon kasvutrendi on tiedostettu myös päättäjien taholla. Thümmeler [Thü15, 15] toteaa, että lähitulevaisuuden tavoite onkin organisoida yhä suurempi osuus terveydenhuoltopalveluista mobiiliterveydenhuoltona. Tällöin potilaat keräävät ja seuraavat itse terveydentilaansa, lääkityksiinsä ja hyvinvointiinsa liittyvää dataa mobiiliterveydenhuollon sovelluksien ja työkalujen avulla. Lisäksi he voivat jakaa tarvittavat tiedot sähköisesti terveydenhuoltohenkilöstölle, jotka voivat antaa tietojen perusteella hoito-ohjeistuksia. Näin perinteistä kasvotusten tapahtuvaa vastaanottokäyntiä ei välttämättä tarvita.

Thümmeler [Thü15, 13, 15] nostaa esille muitakin keskeisiä 2000-luvun innovaatioita sähköisen terveydenhuollon alueella. Merkittäviä edistysaskeleita ovat esimerkiksi pillerinä nieltävät kamerakapselit, leikkaussalirobotiikka, tähystysleikkaukset sekä keinoelimet, kuten sisäkorvaistutukset, raajaproteesit ja insuliinipumput. Niiden avulla potilaita pystytään diagnosoimaan ja heidän terveydentilaansa kohentamaan aivan uudella tavalla. Samalla on kuitenkin huomioitava, että potilaiden terveys ja jopa elämä on ennistä useammin jonkin turvallisuuskriittisen tietoteknisen laitteen tai tietojärjestelmän

varassa. Tällaisten turvallisuuskriittisten järjestelmien toimintahäiriöillä voi olla hyvin-kin ikävät seuraukset, jotka eivät ole aina edes rahalla korvattavissa. Sähköisen terveydenhuollon tietojärjestelmien kehittämistyössä tuleekin panostaa erityisen paljon vaatimusmäärittelyyn sekä suunnitteluun ja testaamiseen, jotta markkinoille julkaistavat tuotteet olisivat turvallisia ja toimivia [Kni02, 549-550].

Edellä lueteltujen edistysaskeleiden ohella 2000-luvulla on alettu panostaa sähköisen terveydenhuollon alueella myös potilaiden tietojen ja potilasasiakirjojen saattamiseen sähköiseen muotoon [Thü15, 13, 15]. Sähköiset potilastietojärjestelmät ovat alkaneet yleistyä, ja potilaskohtaiset paperikansiot väistyvät. Sensitiivisen, yksittäisten henkilöiden terveydentilaa koskevan tiedon siirtyminen lukituista arkistokaapeista sähköisiin tietokantoihin, tietoverkkoihin ja sovelluksiin on tuonut omat haasteensa terveydenhuollon sähköistyessä. Vaikka tiedon tallentaminen, hakeminen, käsittely ja jakaminen ovat toisaalta helpottuneet, on samanaikaisesti jouduttu kiinnittämään aivan erityistä huomiota tietoturvaan ja -suojaan.

Brost ja Hoffmann [BrH15, 133, 135-136] korostavat, että tietojen suojaukseen tulee ottaa kantaa jo tulevan järjestelmän vaatimusmäärittely- ja suunnitteluvaiheessa. He esittelevät monitahoisen turvakonseptin (engl. multilateral security), jossa kaikki mahdolliset järjestelmän käyttäjät ja hyödyntäjät nähdään potentiaalisina turvallisuusuhkina. Uhkista tehdään riskianalyysi ja sen pohjalta asetetaan tietoturvaan ja tietosuojaan liittyvät tavoitteet esimerkiksi luottamuksellisuuden, eheyden, saatavuuden, yhdisteltävyyden ja läpinäkyvyyden suhteen. Kun tavoitteet on asetettu, niiden pohjalta voidaan määrittää vaatimuksia, joiden tulee toteutua tulevassa järjestelmässä.

Thümmler toteaa, että sähköisen terveydenhuollon kehitystyö jatkuu edelleen kiihtyvällä vauhdilla [Thü15, 13]. Eräiden arvioiden mukaan vuonna 2017 maailmassa tullaan käyttämään noin 18 miljardia euroa terveydenhuoltoalan ICT-kustannuksiin. Tulevaisuudessa tullaan varmasti näkemään uusia terveydenhuoltoalaa edistäviä innovaatioita esimerkiksi biometriikkaan, kvanttilaskentaan ja 3D-tulostukseen liittyen. Myös Maailman terveysjärjestö WHO (World Health Organization) on pyrkinyt toimillaan edesauttamaan sitä, että päättäjät ja valtiot ympäri maailmaa alkaisivat panostaa enemmän sähköisen terveydenhuollon kehittämiseen [Bui06]. Vuonna 2006 julkaistussa raportissaan se kehotti jäsenmaitaan laatimaan pitkän aikavälin strategiasuunnitelmia sähköisen ter-

veydenhuollon ja siihen perustuvien terveydenhuoltopalveluiden kehittämiseksi ja käyttöönottamiseksi. Samalla WHO lupautui tukemaan kehitystä esimerkiksi tarjoamalla sähköiseen terveydenhuollon laatukriteeristön, johon jäsenvaltioiden tulee sitoutua.

Myös Euroopassa on otettu merkittäviä askeleita sähköisen terveydenhuollon edistämiseksi. Euroopan komissio on julkaissut sähköisen terveydenhuollon toimintasuunnitelman vuosille 2012-2020 [Säh12, 7-15]. Toimintasuunnitelmassa asetetaan tavoitteita, kuten sähköisen terveydenhuollon tietojärjestelmien entistä paremman yhteentoimivuuden saavuttaminen, niiden käyttöönoton helpottaminen ja nykyistä laajemman käytön varmistaminen. Käytännön toimina esitetään muun muassa yhteisten standardien kehittämistä, lainsäädännön yhtenäistämistä ja tiivistä maiden välistä yhteistyötä. Yksi konkreettinen esimerkki tästä on eurooppalainen epSOS-hanke (Smart Open Services for European Patients) [Mik14]. Sen puitteissa kokeiltiin potilastietojen, kuten sähköisten lääkemääräysten, toimittamista yli maarajojen. Hanke päättyi vuonna 2014, mutta sen rohkaisevia tuloksia tultaneen hyödyntämään sähköisen terveydenhuollon tietojärjestelmien standardoinnissa ja ohjeistuksissa.

Purtova, Kosta ja Koops [PKK15, 48-49] tuovat esille, että sähköisen terveydenhuollon tietojärjestelmiä kuten terveydenhuoltoalaa yleensäkin määrittävät lukuisat lait ja säädökset. Joissakin tapauksissa lait tai asetukset voivat liittyä suoraan sähköisen terveydenhuollon tietojärjestelmään, kuten laki sähköisestä lääkemääräyksestä [Lak14a]. Jotkut laeista ja asetuksista taas liittyvät terveydenhuoltoon ja vaikuttavat sitä kautta myös sähköiseen terveydenhuoltoon. Esimerkiksi Sosiaali- ja terveysministeriön asetus lääkkeen määräämisestä [Sos10, 2. luku] määrittelee, kuka voi määrätä lääkettä. Tietojärjestelmä on rakennettava siten, että tällaisetkin lait ja asetukset toteutuvat. Lisäksi osa laeista määrittää jotakin asiaa tai ominaisuutta, joka ilmenee sähköisen terveydenhuollon tietojärjestelmässä. Tällaisia asioita ovat esimerkiksi tietojärjestelmässä käsiteltävät tiedot, joihin liittyy erinäisiä lakeja ja asetuksia tietosuojaan tai potilaiden yksityisyyden suojaan liittyen. Vastaavasti sähköisen terveydenhuollon tietojärjestelmätuotteen, kuten insuliinipumpun ominaisuuksia saatetaan määrittää esimerkiksi tuoteturvallisuuteen liittyvissä laeissa. Purtovan, Koston ja Koopsin [PKK15, 48-49] mukaan sähköisen terveydenhuollon tietojärjestelmien kehitystyössä tulee aina selvittää paikallinen lainsäädäntö ja huomioida sieltä nousevat vaatimukset.



Volk, Falk-Andersson ja Sedlar [VFS15, 157-159] ovat havainneet, että sähköisen terveydenhuollon tietojärjestelmiin liittyy usein lukuisia sidosryhmiä. Sidosryhmät voivat koostua esimerkiksi kliinisestä henkilökunnasta, potilaista, päättäjistä ja lainsäätäjistä. Usein heidän toiveensa ja tarpeensa tulevalle järjestelmälle ovat hyvin erilaisia ja jopa ristiriitaisia. Vaatimusmäärittelyn tarkoituksena on koota nämä sidosryhmät yhteen ja saada selville yhteinen näkemys siitä, miten tulevan järjestelmän toivotaan toimivan. Vaatimusmäärittelytyön aluksi sidosryhmät kannattaa priorisoida heidän asemansa ja tärkeytensä mukaan, jolloin keskeisimmässä asemassa olevilla sidosryhmillä on suurempi painoarvo. Usein tällaisia keskeisiä sidosryhmiä ovat tilaajataho ja resursoinnista tai rahoituksesta vastaavat tahot.

Kuten edellä esitettiin, sähköisen terveydenhuollon tietojärjestelmillä on tiettyjä erityispiirteitä. On mahdollista, että tarkempi analyysi paljastaisi vielä muitakin erityispiirteitä, mutta tässä tutkielmassa keskeisiksi erityispiirteiksi tunnistettiin seuraavat seikat: sähköisen terveydenhuollon tietojärjestelmiltä edellytetään yhteentoimivuutta muiden toimialan ja organisaatioiden tietojärjestelmien kanssa, ne ovat luonteeltaan usein turvallisuuskriittisiä, niissä käsiteltävien sensitiivisten tietojen vuoksi tietosuojaan tulee panostaa erityisesti, ja niitä säätelevät lait ja asetukset. Lisäksi niihin liittyy useita sidosryhmiä omine intresseineen ja tarpeineen. Nämä erityispiirteet tulee huomioida tietojärjestelmien kehitystyössä jo vaatimusmäärittelyvaiheesta lähtien.

### ***3.2 Sähköinen lääkemääräys ja sen hyödyt ja ongelmat***

Tässä tutkielmassa tutustutaan sähköiseen lääkemääräykseen yhtenä sähköisen terveydenhuollon esimerkkitapauksena. Lääkemääräyksellä eli reseptillä tarkoitetaan lääkkeen määrääjän antamaa määräystä, jonka mukaisesti apteekki toimittaa lääkkeen asiakkaalle [Kel14, 105 ja Näi07]. Lääkemääräys sisältää muun muassa potilastietoja, määrättävän lääkkeen nimen, lääkemuodon, vahvuuden, käyttöaiheen sekä hoidon keston. Lääkäri kirjoittaa em. tiedot paperille joko käsin tai koneella ja antaa lääkemääräyksen potilaan mukaan vietäväksi apteekkiin. Apteekki toimittaa lääkemääräyksen, ja potilas saa siinä määritellyt lääkkeet mukaansa. Joissakin tapauksissa lääkäri voi määrätä lääkkeet ottamalla yhteyttä suoraan apteekkiin puhelimitse tai faksilla. Nykyään lääkärin tulee ensisijaisesti kirjoittaa lääkemääräys sähköisessä muodossa eli e-reseptinä. Sähköisellä lääkemääräyksellä tarkoitetaan tietokonepohjaisesti luotua, siirrettyä ja toimitettua lääke-

määräystä. Se sisältää samat tiedot kuin paperimuotoinen lääkemääräyskin, ja turvallisudellaan, nopeudellaan sekä vaivattomuudellaan se on korvaamassa paperi-, puhelin- ja faksireseptit [Acl08, 1].

Ridinger [Rid07, 13] tuo esille, että pelkästään vuonna 2007 Yhdysvalloissa aiheutui noin 37 miljardin kustannukset lääketieteellisistä virheistä. Nimenomaan lääkityksestä aiheutuvat virheet johtivat jopa 7000 ihmisen kuolemaan ja noin kahden miljardin dollarin sairaalakuluihin. Ainakin osa näistä ikävistä virheistä olisi voitu välttää sähköisten lääkemääräyksien avulla. Merkittävimpiä sähköisen lääkemääräyksen hyötyjä onkin potilasturvallisuuden parantaminen. Sähköisen lääkemääräyksen tietojärjestelmään voidaan integroida päätöksentekijärjestelmä, joka huomioi asiakkaan kokonaistilanteen tuoden esille mahdolliset lääkeaineallergiat sekä lääkkeiden yhteisvaikutukset [PEC14, 3 ja Rid07, 15]. Näin lääkkeiden määrääjä pystyy valitsemaan juuri kyseiselle asiakkaalle soveltuvia lääkkeitä ja virheiden mahdollisuus pienenee. Useimmat järjestelmät myös tunnistavat poikkeukselliset annostukset, jolloin ne pyytävät lääkkeen määrääjää tarkistamaan, onko todella tarkoitus määrätä aiottu annostus potilaalle. Tämän lisäksi potilasturvallisuutta parantaa se, että apteekin työntekijöiden on helpompi tulkita sähköiseen määrämuotoon kirjoitettua lääkemääräystä kuin kiireessä omalla käsialalla kirjoitettua lääkemääräystä. Sähköisen lääkemääräyksen on myös arvioitu vaikeuttavan ja vähentävän lääkemääräykseen liittyviä väärinkäytöksiä, sillä asiakkaan on lähes mahdotonta väärentää lääkemääräystä.

Porterfieldin, Engelbertin ja Coustassen sekä Ridingerin [PEC14, 4 ja Rid07, 14] mukaan toinen olennainen hyöty on sähköisen lääkemääräyksen tehokkuus verrattuna käsin kirjoitettaviin lääkemääräyksiin. Tutkimusten mukaan lääkkeen määrääjältä kuluu lääkemääräyksen kirjoittamiseen lähes sama aika sähköisesti kuin käsin, mutta oheis- ja jälkitöiden määrä on sähköisessä lääkemääräyksessä huomattavasti pienempi. Kun järjestelmä ehdottaa soveltuvia lääkkeitä ja niiden annostuksia, ei määrääjän tarvitse etsiä tietoa manuaalisesti erikseen. Apteekissa työntekijöiltä ei kulu myöskään aikaa tekstien kopioimiseen paperireseptiltä käsin tietojärjestelmään. Lisäksi apteekista joudutaan usein soittamaan käsin kirjoitettuihin lääkemääräyksiin liittyviä tarkistuksia lääkkeen määrääjältä puhelimitse. Eräiden arvioiden [Rid07, 14] mukaan jopa kolmannes käsin kirjoitetuista lääkemääräyksistä edellytti yhteydenottoa lääkkeen määrääjään ja lääkemääräyksen tarkennusta eli käytännössä turhaa työtä. Kokonaisuudessaan on arvioitu,

että sähköiset lääkemääräykset tuottavat Yhdysvalloissa jopa 27 miljardin dollarin vuotuiset säästöt [PEC14, 7].

Kolmas hyötynäkökulma on asiakkaan kokema taloudellinen hyöty [PEC14, 4]. Tämä toteutuu osin sen ansiosta, että järjestelmä auttaa lääkkeen määrääjää löytämään helpommin asiakkaalle sopivan lääkityksen, jolloin säästytään turhilta kokeiluilta. Lisäksi se, että järjestelmä ehdottaa automaattisesti mahdollisia edullisempia rinnakkaislääkkeitä, auttaa osaltaan pienentämään asiakkaan lääkekuluja. Asiakas hyötyy myös siitä, että lääkemääräysten tiedot siirtyvät sähköisesti suoraan apteekkiin, jolloin hänen ei tarvitse huolehtia paperireseptien säilyttämisestä.

Vaikka sähköisen lääkemääräyksen hyödyt ovat kiistämättömiä, sen käyttöönottoon ja käyttöön liittyy myös tiettyjä ongelmia, kuten hankinta- ja ylläpitokustannukset [PEC14, 6]. Uusi tietojärjestelmä vaatii toimiakseen soveltuvat laitteistot, ohjelmistot ja integraation muihin osajärjestelmiin. Tällainen investointi voi muodostua hyvin kalliiksi, varsinkin jos kyseessä on pieni toimija, joka joutuu kustantamaan kaikki tarvittavat elementit yksin. Suurilla ketjuilla kustannukset jakautuvat usein laajemmalle eivätkä vaikuta organisaation talouteen vakavasti. Kaikki tietojärjestelmät vaativat myös ylläpitoa, eli järjestelmän toimivuus edellyttää hankintakustannusten lisäksi vuosittaista budjetointia ylläpitokustannuksiin.

Ridingerin [Rid07, 15] mukaan järjestelmän käyttöönotossa ongelmaksi saattaa muodostua myös henkilökunnan osaamisvaje, motivaatio-ongelmat tai muutosvastarinta. Olennaista on, että osaamisen kasvattamiseen ja motivointiin panostetaan jo etukäteen, jotta käyttöönoton yhteydessä suurimmat puutteet on jo saatu korjattua. Uuden tietojärjestelmän käyttöönotto edellyttää usein myös työtapojen ja -prosessien uudelleen organisointia, joten niitä on työstettävä yhdessä henkilökunnan kanssa.

Sähköisen lääkemääräyksen osajärjestelmissä käsitellään paljon hyvin arkaluonteista tietoa, joka ei saa päätyä aiheettomasti ulkopuolisten tietoon [PEC14, 6 ja Rid07, 14]. Järjestelmän käyttäjien ja asiakkaiden luotettava tunnistaminen on tämän vuoksi erittäin tärkeää. Kaikkien järjestelmän käyttäjien tulee sitoutua sovittuihin tietoturvamenettelyihin, esimerkiksi omia käyttäjätunnuksia ei saa lainata työtoverille. Lisäksi organisaation tekninen tietoturva on saatettava kuntoon; palomuurit ja virusturvaohjelmistot on pidet-

tävä ajan tasalla. Jos näitä henkilökuntaan ja fyysiseen ympäristöön liittyviä osatekijöitä ei huolehdita kuntoon, on riski tietojen vuotamiseen suuri.

Tietojärjestelmät ovat aina alttiita häiriöille ja käyttökatkoille, ja tämä koskee myös sähköisiä lääkemääräysjärjestelmiä. Porterfieldin, Engelbertin ja Coustassen [PEC14, 6] havaintojen mukaan ongelmat sähköisen lääkemääräysjärjestelmän ohjelmistossa oli suurin yksittäinen syy (34 %), jonka takia järjestelmän käyttö lopetettiin organisaatiossa. Jotta käyttöä hankaloittavia ongelmia ei ilmenisi, on järjestelmän vaatimusmäärittelyyn ja suunnitteluun kiinnitettävä erityistä huomiota. Lisäksi tulee miettiä menettelyt, kuinka toimitaan silloin, kun järjestelmä ei häiriötilanteessa toimi oletetulla tavalla.

Sähköinen lääkemääräys on hyvä esimerkki sähköisen terveydenhuollon tietojärjestelmistä ja niissä ilmenevistä erityispiirteistä. Turvallisuuskriittisyys ilmenee siinä, että järjestelmän avulla potilaille määrätään tiettyjä lääkkeitä tietyillä annostuksilla. Jos järjestelmähäiriö vaikkapa muuttaisi lääkkeen toiseksi tai tuplaisi annostuksen, se voisi johtaa vakaviin seurauksiin. Tietosuojan tarve korostuu siksi, että sähköisessä lääkemääräysjärjestelmässä käsitellään potilaiden terveystietoja. Olisi hyvin kiusallista, jos ulkopuoliset pääsisivät käsiksi järjestelmän tietoihin ja julkaisisivat vaikkapa jonkun valtion päämiehen lääkitystiedot. Yhteentoimivuutta tarvitaan esimerkiksi tietojen siirtämiseksi moninaisten potilas- ja apteekkitietojärjestelmien kesken. Lakiperusteisuus ilmenee siinä, että järjestelmän tulee noudattaa lakia sähköisestä lääkemääräyksestä sekä muita keskeisiä terveydenhuoltoalan lakeja. Lisäksi järjestelmällä on useita sidosryhmiä aina järjestelmän käyttäjätahoista (lääkärit ja apteekkarit) lainsäätäjiin ja lääkeyrityksiin.

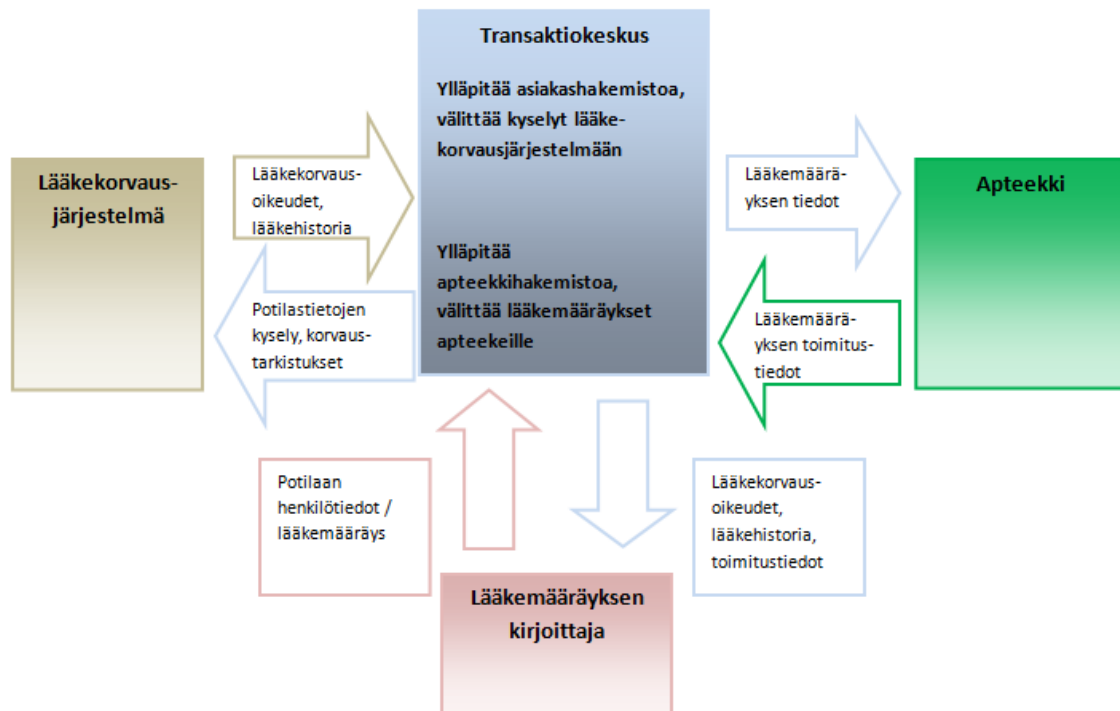
### ***3.3 Sähköisen lääkemääräyksen tietojärjestelmäratkaisut***

Sähköinen lääkemääräys on yleistynyt nopeasti ympäri maailmaa. The eHealth Initiative Foundationin eli EHI:n [Acl08, 1-2] mukaan tarjolla olevat tietojärjestelmäratkaisut eroavat toisistaan suuresti ominaisuuksiltaan ja arkkitehtuuriltaan. Osa järjestelmistä on itsenäisiä (engl. stand-alone), ja osa on toteutettu integraatioina potilastietojärjestelmään. EHI [Acl08, 1, 8-9] on määritellyt toiminnallisuudet, jotka sähköiseen lääkemääräysjärjestelmän tulee voida suorittaa. Tällä on pyritty sähköisen terveydenhuollon yhtenäiseen kehittämiseen sekä potilasturvallisuuden parantamiseen.

EHl:n [Acl08, 1, 8-9] tavoitteena on rakentaa mahdollisimman kattava potilastietojärjestelmä. Määritysten mukaan järjestelmän tulee esimerkiksi tarjota täydellinen lista lääkkeistä, joita järjestelmän kautta voidaan määrätä. Järjestelmän avulla tulee voida valita lääkkeitä, muodostaa niihin pohjautuvia lääkityksiä, tulostaa lääkemääräyksiä ja siirtää lääkemääräyksiä sähköisessä muodossa. Järjestelmän on esitettävä mahdolliset hinnaltaan edullisemmat rinnakkaislääkevaihtoehdot. Järjestelmän tulee tehdä automaattisia turvatarkastuksia muun muassa lääkkeiden annostuksiin, yhteisvaikutuksiin ja allergioihin liittyen. Lisäksi järjestelmän tulee tarjota tietoa lääkkeiden korvattavuuksista ja asiakkaan korvausoikeuksista.

Sähköinen lääkemääräyksen tietojärjestelmäratkaisu vaatii toimiakseen erilaisia yhteentoimivia komponentteja ja niitä yhdistäviä prosesseja [How14, Ele15]. Yhdessä ne muodostavat tietojärjestelmän, joka mahdollistaa sähköisten lääkemääräysten kirjoittamiseen liittyvät toiminnot sekä määräysten mukaisten lääkkeiden toimittamisen apteekista. Yhdysvaltalainen The Health Resources and Services Administration (jatkossa HRSA) on mallintanut sähköisen lääkemääräyksen koostuvan neljästä komponentista: lääkemääräyksen kirjoittajasta (engl. prescriber), apteekista (engl. pharmacy), lääkekorvausjärjestelmästä (engl. pharmacy benefit manager eli PBM) sekä näitä yhdistävästä transaktiokeskuksesta (engl. transaction hub). Komponentit ja niiden väliset prosessit on esitelty kuvassa 7.

Kuvassa 7 esitetyssä mallissa jokaisella komponentilla on tietyt tehtävät ja proseduurit, joiden mukaan ne ovat yhteydessä toisiinsa [How14]. Ensin lääkemääräyksen kirjoittaja kirjautuu järjestelmään ja pääsee tunnistautumisen jälkeen työstämään sähköisiä lääkemääräyksiä. Hän hakee tietyn asiakkaan tietoja, kuten henkilötiedot, lääkehistorian ja korvausoikeudet. Kysely välittyy transaktiokeskukseen, jossa säilytetään asiakashakemistoa. Kun lääkemääräyksen kirjoittaja on tunnistanut ja valinnut oikean asiakkaan asiakashakemistosta, transaktiokeskus suorittaa asiakastietojen täydennyskyselyn lääkekorvausjärjestelmään. Lääkekorvausjärjestelmä palauttaa asiakkaan lääkehistoriaan ja lääkekorvausoikeuksiin liittyvät tiedot transaktiokeskukseen, joka puolestaan välittää tiedot lääkemääräyksen kirjoittajalle.



Kuva 7. Sähköisen lääkemääräyksen tietojärjestelmäratkaisun komponentit ja niiden väliset prosessit HRSA:n mukaan [How14 mukaillen].

Tämän jälkeen lääkemääräyksen kirjoittaja voi tutustua asiakkaan lääkehistoriaan, kirjoittaa uuden lääkemääräyksen, muokata jotakin olemassa olevaa lääkemääräystä tai uusia lääkemääräyksen [How14]. Seuraavaksi hän valitsee transaktiokeskuksen ylläpitämästä hakemistosta apteekin, josta lääke tullaan asiakkaalle toimittamaan. Lääkemääräyksen tiedot lähetetään transaktiokeskukseen, joka välittää tiedot eteenpäin. Lääkitystiedot lähetetään lääkekorvausjärjestelmään korvausoikeustarkastusta varten sekä lisättäväksi asiakkaan lääkehistoriaan. Lääkemääräyksen tiedot lähetetään apteekkijärjestelmään odottamaan toimittamista. Tämän jälkeen asiakas menee apteekkiin noutamaan lääkettä. Apteekkijärjestelmä lähettää toimittamisesta tiedot transaktiokeskukselle, joka välittää tiedon lääkemääräyksen kirjoittajan organisaatioon asiakkaan lääkityssuunnitelman toteutumisen seuraamiseksi.

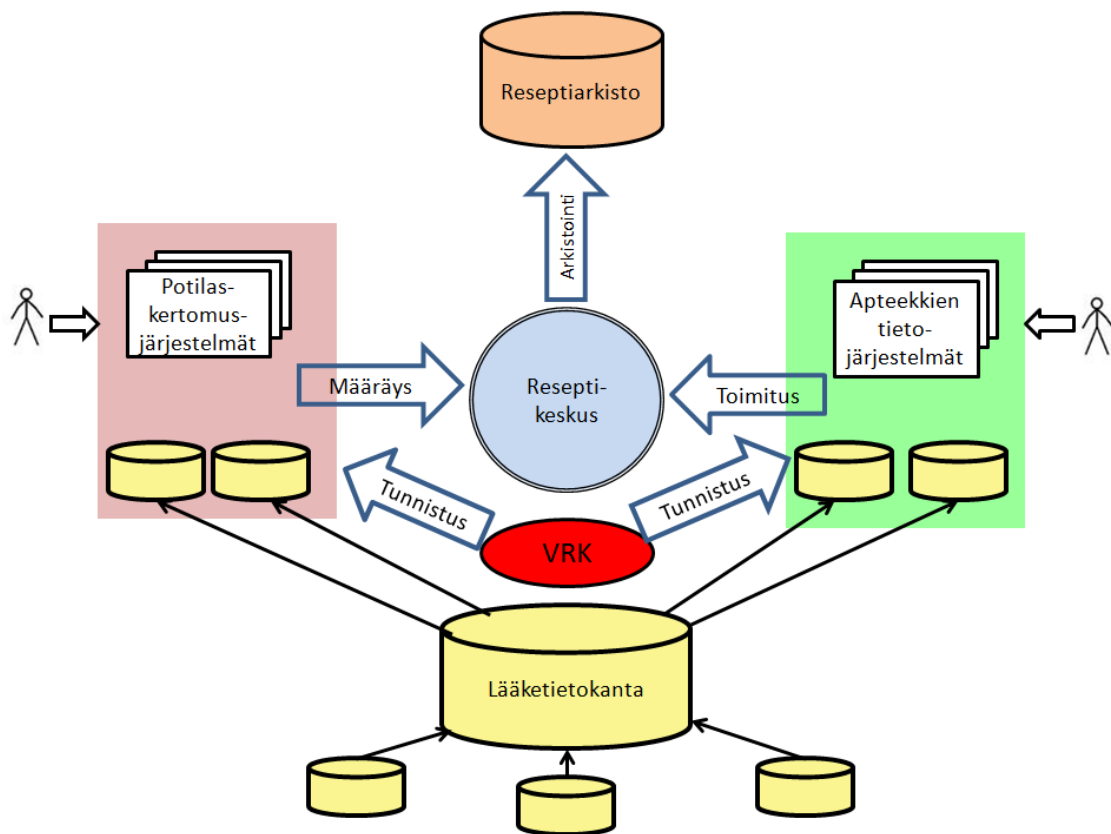
Suomessa sähköinen lääkemääräys ja sen kehittäminen on delegoitu Kansaneläkelaitoksen eli Kelan alaisuudessa toimivan Kansallisen Terveysarkiston (Kanta) vastuulle [Säh15b]. Sähköistä lääkemääräystä säätelevät useat eri lait ja asetukset, joista olennaisin on vuonna 2007 voimaan tullut (61/2007) ja vuosina 2010 (436/2010) sekä 2014 (251/2014) muutettu laki sähköisestä lääkemääräyksestä [Lak14a ja Lak14b]. Laissa

säädetään sähköisen lääkemääräyksen käyttöönotto pakollisiksi apteekkeille, terveydenhuollon toimintayksiköille sekä tällaisissa yksiköissä vastaanottoa pitävälle ammatinharjoittajille. Lain tarkoituksena on potilas- ja lääketurvallisuuden parantaminen sekä lääkkeiden määräämisen ja toimittamisen helpottaminen ja tehostaminen.

Suomen ensimmäinen sähköinen lääkemääräys kirjoitettiin pilotin yhteydessä Turussa vuonna 2010 [Vuo14]. Pilotin jälkeen sähköinen lääkemääräys on yleistynyt vähitellen; vuoden 2012 keväästä lähtien kaikki apteekit ovat olleet e-reseptivalmiudessa, ja tällä hetkellä kaikki sekä julkisen että yksityisen terveydenhuollon yksiköt ovat sen piirissä. Sähköisen lääkemääräyslain [Lak14a] mukaan tulee myös omissa toimitiloissaan toimivien itsenäisten ammatinharjoittajien sekä Ahvenanmaan maakunnassa toimivien palveluntarjoajien ryhtyä kirjoittamaan lääkemääräykset sähköisinä vuoden 2016 loppuun mennessä. Sähköisen lääkemääräyksen käyttö edellyttää asiakkaan suostumusta, ja hänellä on vuoden 2016 loppuun asti mahdollisuus saada halutessaan lääkemääräys myös paperimuodossa, puhelimitse tai faksilla. Lisäksi lääkemääräys voidaan antaa muussa kuin sähköisessä muodossa mahdollisten käyttökatkojen aikana, jotta potilasturvallisuus ei vaarantuisi.

Suomen sähköisen lääkemääräyksen rakenne noudattelee osin aiemmin esiteltyä HRSA:n määrittelemää ratkaisua, mutta siinä on omat erityispiirteensä. Kuten kuvasta 8 ilmenee, Suomen sähköisen lääkemääräysjärjestelmään kuuluu seuraavat komponentit: potilaskertomusjärjestelmät, apteekkien tietojärjestelmät, lääketietokanta, reseptikeskus ja reseptiarkisto [Säh15a, 9, 15, 22].

Kuvan 8 mukaisesti tietojärjestelmän käyttö tapahtuu lääkemääräyksiä kirjoitettaessa potilaskertomusjärjestelmien ja lääkemääräysten mukaisia lääkkeitä toimitettaessa apteekkien tietojärjestelmien kautta [Säh15a, 10]. Rakenne on samantyyppinen, kun edellä esitellyssä HRSA:n ratkaisussa, jossa apteekkeilla ja lääkkeen määrääjillä on omat roolinsa ja toiminnallisuutensa. Syksyllä 2014 eHealth-seminaarityöhön haastateltu Kanta-asiantuntija Katriina Köli [Köl14] kertoi, että haastatteluhetkellä tietojärjestelmään liit-  
tyneitä potilaskertomusjärjestelmiä oli 15 kappaletta ja apteekkijärjestelmiä kolme.



Kuva 8. Suomen sähköisen lääkemääräyksen tietojärjestelmäratkaisu [Säh15a, 20 mukaillen].

Kölin [Köl14] mukaan sähköisen lääkemääräyksen suunnittelun yhteydessä järjestelmiä ei siis pyritty yhdistämään yhdeksi järjestelmäksi. Sen sijaan päädyttiin integroimaan olemassa olevat potilas- ja apteekkijärjestelmät osaksi uutta tietojärjestelmää. Edellytyksenä integraatiolle olivat toimivat rajapinnat, joiden kautta osajärjestelmät ovat yhteydessä muihin komponentteihin. Rajapinnat on toteutettu käyttämällä yleisiä standardeja, kuten HL7 v3 ja CDA R2 [Säh15a, 23]. Lisäksi edellytyksenä on, että kaikki osapuolet käyttävät Terveiden ja hyvinvoinnin laitoksen (THL) ja HL7 Finland ry:n toteuttamia koodistoja. Kaikkien järjestelmän käyttäjien on tunnistauduttava, joka tapahtuu Väestörekisterikeskuksen (VRK) kautta [Säh15a, 9].

Sähköiset lääkemääräykset ja lääkkeiden toimitustiedot tallennetaan Reseptikeskukseen [Säh15a, 15-16]. Se toimii ratkaisussa vain tietokantana, eli sillä ei ole varsinaisia toiminnallisuuksia eikä omaa käyttöliittymää. Tämä noudattelee HRSA:n laatimaa ratkaisua, jossa transaktiokeskus välittää tietoja eri komponenttien välillä. Sen sijaan HRSA:n ratkaisusta poiketen Reseptikeskus ei hallinnoi asiakashakemistoa, ei suorita määräyk-



sille ja toimituksille tietosisältötarkastuksia eikä suorita käyttäjien tunnistusta, vaan se operoi potilaskertomus- ja apteekkijärjestelmien tekemien tunnistusten perusteella. Lääkemääräykset säilyvät Reseptikeskuksessa 30 kuukautta, minkä jälkeen ne siirtyvät Reseptiarkistoon, jossa niitä säilytetään 10 vuotta. Arkistointikäytäntö tulee kuitenkin muuttumaan 1.1.2017. Sen jälkeen uusia lääkemääräyksiä ei enää arkistoida, vaan sähköiset lääkemääräykset ovat voimassa oletuksena 24 kuukautta määräämispäivämäärästä lukien.

Järjestelmään kuuluu myös Lääketietokanta, jonne kerätään tietoja muista lähdejärjestelmistä [Säh15a, 22-23]. Sinne tallennetaan esimerkiksi lääketietokanta Lääketietokeskukselta, hintatiedot lääkeyrityksiltä ja korvattavuustiedot Lääkkeiden hintalautakunnalta. Lääketietokannan tiedot päivitetään kahden viikon välein potilaskertomusjärjestelmiin ja apteekkien tietojärjestelmiin.

### **3.4 Suomen sähköisen lääkemääräyksen toiminnallisuudet**

Sähköinen lääkemääräysjärjestelmä tarjoaa erilaisia toiminnallisuuksia käyttäjilleen [Säh15a, 16-22]. Lääkkeiden määrääjät tai asiakkaiden hoitohenkilökunta käyttävät siis järjestelmää omien organisaatioidensa potilaskertomusjärjestelmien kautta ja apteekkien henkilökunta apteekkien tietojärjestelmien kautta. Käyttäjät voivat suorittaa erilaisia toiminnallisuuksia riippuen heille annetuista oikeuksista. Osa seuraavaksi esiteltävistä toiminnallisuuksista on suoritettavissa vain potilastietojärjestelmien kautta ja osa vain apteekkien tietojärjestelmien kautta. Joitakin toiminnallisuuksia voidaan suorittaa sekä potilastietojärjestelmien että apteekkien tietojärjestelmien kautta. Liitteeseen 2 on koottu yhteenveto keskeisimmistä sähköisen lääkemääräysjärjestelmän tarjoamista toiminnallisuuksista ja siitä, mitä kautta niitä voidaan suorittaa.

Kuten liitteestä 2 ilmenee, potilastietojärjestelmien kautta sähköisen lääkemääräyksen käyttäjät voivat tehdä useita lääkemääräyksiin kohdistuvia toimenpiteitä [Säh15a, 16-21]. Lääkäri tai muu valtuutettu hoitohenkilökunta voi esimerkiksi *katsella* potilaan lääkemääräystietoja ja -historiaa. Potilaan lääkemääräys voidaan hakea Reseptikeskuksesta henkilötunnuksen ja yksilöivän lääkemääräyksen tunnisteen avulla. Käyttöoikeuksien puitteissa potilastietojärjestelmään voidaan myös kirjata uusi sähköinen *lääkemääräys* eli määrätä potilaalle tämän tarvitsemaa lääkettä. Jos lääkemääräyksessä on jokin virhe, voidaan määräystä tarvittaessa *korjata*. Tällöin korjauksesta muodostuu järjestel-

mään uusi versio lääkemääräyksestä, ja aiempi versio jää järjestelmän versiohistoriaan. Potilastietojärjestelmän kautta sähköinen lääkemääräys voidaan myös *mitätöidä*, jos se todetaan tarpeettomaksi. Myös mitätöinti synnyttää järjestelmään uuden version vanhan säilyessä versiohistoriassa. Jos lääkemääräys on apteekkihenkilökunnan toimesta lukittu esimerkiksi jonkin yksityiskohdan selvittämistä varten, *lukituksen purku* voidaan tehdä potilastietojärjestelmän kautta. Jos potilas haluaa, että hänen lääkemääräyksensä uusi-taan, siitä voidaan kirjata *uusimispyyntö* potilastietojärjestelmän kautta. Myös lääke-määräyksen *uusiminen* tai tarvittaessa *uudistamiskiellon asettaminen* tehdään potilastie-tojärjestelmästä käsin.

Potilastietojärjestelmien kautta hoidetaan myös erinäisiä potilaan informointiin ja tie-donkulkuun liittyviä asioita [Säh15a, 21-22]. Sieltä voidaan esimerkiksi tulostaa poti-laalle uuteen lääkemääräykseen liittyvä *potilasohje* sekä *yhteenvedo* hänen lääkemää-räyksistään ja niiden toimituksista. Jos potilas aikoo ostaa lääkkeitä ulkomailta, hänelle voidaan tulostaa lääkemääräyksestä *jäljennös ulkomaanostoa varten*.

Liitteen 2 mukaisesti myös apteekkien tietojärjestelmien kautta on tarjolla useita toi-minnallisuuksia liittyen joko sähköiseen lääkemääräykseen tai sen toimitukseen. Far-maseutit tai muut lääkemääräysten toimittamiseen valtuutetut henkilöt voivat *katsella* asiakkaan lääkemääräystietoja ja -historiaa [Säh15a, 19-20]. He voivat myös tarvittaessa käyttöoikeuksiensa puitteissa *korjata lääkemääräystä* lääkkeen määrääjän luvalla; tosin pienet tekniset korjaukset eivät edellytä lääkkeen määrääjän suostumusta. Samoin ai-heettomaksi todetun *lääkemääräyksen mitätöinti* onnistuu apteekkitietojärjestelmän kautta. Sekä lääkemääräyksen korjaus että mitätöinti tuottavat myös apteekkijärjestel-mien kautta tehtynä uuden version lääkemääräyksen versiohistoriaan.

Jos apteekkihenkilöstö epäilee lääkemääräyksen olevan virheellinen, heidän on voitava suorittaa *lääkemääräyksen lukitus* [Säh15a, 19-20]. Lukitus estää lääkkeen toimittami-sen, ja virhe-epäilyä voidaan selvittää ottamalla yhteyttä lääkkeen määrääjään. Lääke-määräys on lukittuna niin kauan, kunnes *lukitus puretaan* joko potilas- tai apteekkitieto-järjestelmän kautta. Apteekkitietojärjestelmän kautta voidaan asiakkaan pyynnöstä suo-rittaa *lääkemääräyksen varaus* ja *varauksen purku*. Varattu lääkemääräys voidaan toi-mittaa ainoastaan varauksen suorittaneesta apteekista. Jos lääkemääräys kohdistuu eri-tyisluvalliseen valmisteeseen, tekee apteekki *lääkemääräyksen erityislupavarauksen*.

Myös tällöin lääkemääräyksen toimittaminen on mahdollista ainoastaan erityislupavarauksen tehneestä apteekista. Tarvittaessa voidaan myös suorittaa *erityislupavarauksen purku*.

Kuten liitteestä 2 ilmenee, apteekkien tietojärjestelmien kautta on mahdollisuus tehdä useita lääkemääräyksen toimitukseen liittyviä toimenpiteitä [Säh15a, 19-20]. Apteekkihenkilöstö voi käyttöoikeuksiensa puitteissa esimerkiksi suorittaa *lääkemääräyksen toimituksen* eli myydä asiakkaalle lääkemääräyksen mukaisen lääkkeen. Tällöin kyseinen lääkemääräys asetetaan *toimitusvaraus*-tilaan, eikä sitä voida samanaikaisesti toimittaa muista apteekeista. Toimitusvaraus purkautuu automaattisesti, jos lääkemääräykseen tallentuu toimitus tai jokin muu toimenpide. Jos toimitusta tai muuta toimenpidettä ei suoriteta, voi apteekki *purkaa toimitusvarauksen*. Mikäli toimitusvaraus kuitenkin jää jostain syystä aktiiviseksi, poistaa Reseptikeskuksen hallintajärjestelmä päälle jääneet toimitusvaraukset kerran vuorokaudessa. Lääkemääräyksen toimituksen yhteydessä asiakkaalle tulostetaan *lääkemääräyksen toimitustieto*. Kyseessä on lääkepakkauksen kylkeen liimattava tarra, josta selviävät lääkkeen toimitustiedot, kuten apteekin yhteystiedot, lääkkeen toimituksen päivämäärä, määräyksen kirjoittaneen lääkärin nimi. Tarvittaessa apteekkien tietojärjestelmistä voidaan myös *korjata tai mitätöidä lääketoimitus*, jolloin uusi tieto tallentuu järjestelmiin uutena versiona vanhan jäädessä talteen versiohistoriaan.

Joissakin tapauksissa asiakas saattaa pyytää hänelle määrättyt lääkkeet pakattavaksi apteekissa valmiiksi esimerkiksi päiväannoksiin [Säh15a, 17-20]. Tällöin apteekkitietojärjestelmään tallennetaan tieto *annosjakelusta*. Tarvittaessa voidaan myös suorittaa *annosjakelun purku*, mikäli asiakas sitä pyytää. Jos asiakas on lähdössä ulkomaille ja tarvitsee mukaansa kirjallisen todistuksen hänelle toimitetuista lääkkeistä (esimerkiksi huumauslääkeaineet), hänelle voidaan apteekkitietojärjestelmästä tulostaa *lääkemääräyksestä jäljennös* allekirjoituksella vahvistettuna. Asiakkaalle voidaan myös tulostaa *yhteenveto* hänen lääkemääräyksistään ja niiden toimitustiedoista. Lisäksi asiakkaan pyynnöstä voidaan kirjata apteekkitietojärjestelmään *lääkemääräyksen uusimispyyntö*. Uusimispyyntöviesti ohjautuu Reseptikeskuksen kautta asiakkaan ilmoittaman terveydenhuollon organisaation potilastietojärjestelmään.

Kaikki edellä esitelty toiminnot edellyttävät määrättyjä käyttöoikeuksia ja vahvaa sähköistä tunnistautumista [Säh15a, 10-12]. Käyttöoikeuksien hallinta tapahtuu potilas- ja apteekkitietojärjestelmissä, joissa käyttöoikeuksien tulee olla määritelty työtehtäväpohjaisesti. Järjestelmien tulee myös tarkistaa henkilöiden oikeudet ja varmenteiden voimassaolo Valviran palvelusta tunnistautumisen yhteydessä. Kaikista suoritetuista toiminnoista jää merkintä lokiin. Lisäksi haettaessa tietoa katseltavaksi Reseptikeskuksesta on haun yhteydessä annettava syy tietojen kyselyyn sekä tieto tarvittavan suostumuksen olemassaolosta. Nämä tallentuvat Reseptikeskukseen, josta niitä voidaan tarvittaessa tutkia esimerkiksi tietojen aiheettoman katselun tai väärinkäyttöepäilyjen yhteydessä. Lääkemääräykset ja -toimitukset sekä näiden korjaukset ja mitätöinnit allekirjoitetaan aina sähköisesti ulkopuolisen allekirjoituspalvelun avulla.

Edellä on esitelty laaja kattaus Suomen sähköisen lääkemääräyksen toiminnallisuuksia. Mistä järjestelmän ohjelmoijat ovat tienneet, mitä toiminnallisuuksia tietojärjestelmään pitää ohjelmoida, jotta se täyttää käyttäjiensä tarpeet? Miten nämä toiminnallisuudet on tunnistettu? Vastaus on vaatimusmäärittelytyö. Kuten tämän tutkielman teoriaosuudessa tuli ilmi, toimivan tietojärjestelmän kehitystyön lähtökohtana on vaatimusmäärittely. Myös sähköisen lääkemääräyksen kehitystyö on aloitettu selvittämällä sidosryhmien tarpeet ja tekemällä vaatimusmäärittelyä.

## **4 Malli sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittelyyn**

Suomen sähköisen lääkemääräyksen vaatimusmäärittelytyöstä on saatavilla jonkin verran julkista materiaalia Kansallisen terveysarkiston IT-ammattilaisille suunnatuilla Internet-sivuilla. Sivuilta löytyy muun muassa sähköisen lääkemääräyksen vaatimusmäärittelyn yleiskuvaus sekä potilastietojärjestelmiin ja apteekkitietojärjestelmiin kohdistuvat vaatimukset. Materiaalista selviävät kyllä tunnistetut vaatimukset ja niihin liittyviä yksityiskohtia, mutta ei tarkempaa kuvausta siitä, miten varsinainen vaatimusmäärittelytyö on suoritettu. Vaatimusmäärittelytyön suorittamisen yksityiskohtia yritettiin selvittää Kanta-asiantuntija Katriina Kõliltä, joka on omalta osaltaan vastaamassa sähköisen lääkemääräyksen vaatimustenhallinnasta. Häntä haastateltiin eHealth-seminaarityöhön syksyllä 2014 [Kõl14]. Kõli kertoi, ettei hän ollut vielä nykyisissä Kanta-tehtävissään,

kun sähköisen lääkemääräyksen vaatimusmäärittelyä suoritettiin. Hänelläkään ei ollut tietoa vaatimusmäärittelytyön suorittamisen yksityiskohdista.

Tätä tutkielmaa tehdessä ilmeni, ettei ole olemassa koestettua mallia tai sapluunaa siihen, miten tällaisten monimutkaisten ja lakiperustaisten, sensitiivisiä tietoja sisältävien, turvallisuuskriittisten sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittelytyö voitaisiin tai olisi suositeltavaa suorittaa. Kehitysprojektit joutuvat siis keksimään vaatimusmäärittelytyönsä käytänteet aina uudelleen, yrityksen ja erehdyksen kautta. Tässä luvussa esitellään tutkielman teoriaosuuteen pohjautuva malli, kuinka sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittelytyö kannattaisi suorittaa. Malli koostuu kolmesta erilaisesta vaatimusmäärittelytyön iteraatiovariaatiosta ja niihin sisältyvistä toiminnoista, jotka esitellään tarkemmin tulevissa alaluvuissa.

Hypoteettisena esimerkkinä käytetään Suomen sähköistä lääkemääräystä ja esitellään, kuinka vaatimusmäärittelymallia olisi voitu soveltaa siihen liittyvässä vaatimusmäärittelytyössä. Taustaoletuksena on, että lääkemääräykset on aiemmin kirjoitettu paperimuodossa, eli vaatimusmäärittely kohdistuu täysin uuteen tietotekniseen järjestelmään. Malliesimerkissä ei sinällään pyritä tunnistamaan tai nimeämään sähköiseen lääkemääräykseen kohdistuvia vaatimuksia, vaikka mahdollisia vaatimusehdokkaita esimerkinomaisesti tuodaankin esille. Sen sijaan mallissa pyritään kuvaamaan, mitä asioita missäkin vaatimusmäärittelyn iteraatiossa ja työvaiheessa olisi hyvä tehdä ja huomioida.

Mallissa sovelletaan van Lamsweerden määrittelemiä vaatimusmäärittelyn työvaiheita ja vaatimusmäärittelyn dokumentaation laatutavoitteita [Lam09, 30-36], Hofmannin ja Lehnerin esittelemiä vaatimusmäärittelyn parhaita käytäntöjä [HoL01, 65-66] sekä El Emamin ja Madhavjin tunnistamia vaatimusmäärittelyn onnistumistekijöitä [EIM95, 208-209]. Mallista on laadittu karkeistettu muistilista liitteeseen 3. Muistilista auttaa hahmottamaan ja muistamaan, millaisiin asioihin missäkin työvaiheessa on tarpeen kiinnittää huomiota.

#### ***4.1 Vaatimusmäärittelyn esivalmistelut***

Ennen vaatimusmäärittelyn aloittamista on hyvä varmistaa tiettyjä asioita, jotta vaatimusmäärittelyllä olisi edellytykset onnistua. Jo projektin esiselvitysvaiheessa laaditaan vaatimusmäärittelyn strategia. Sitä varten on analysoitava, millaisesta tietojärjestelmäs-

tä, kehitysprojektista ja organisaatiosta on kyse. Analyysin pohjalta valitaan vaatimusdokumentaation tavoitteet ja vaatimusmäärittelytyön onnistumistekijät, joihin on erityisesti tarve panostaa. Sähköisen lääkemääräyksen osalta voitaisiin esimerkiksi todeta, että vaatimusmäärittelydokumentaation täydellisyys ja sopivuus ovat muita laatutavoitteita tärkeämpiä. Onnistumistekijöistä saatettaisiin priorisoida vaikkapa liiketoimintaprosessin kuvastuminen arkkitehtuurissa ja vaatimusmäärittelydokumentaatioon tehtävien muutosten määrä. Tehdyt valinnat ohjaavat projektin suunnittelua ja läpivientiä.

Lisäksi ennen varsinaisen vaatimusmäärittelytyön alkamista tulee varmistaa asioita panostukseen ja henkilöresursseihin liittyen. Kuten tutkielman teoriaosuudessa esitettiin, vaatimusmäärittely on avainasemassa toimivan ja asiakkaan tarpeet täyttävän tietojärjestelmän tuottamisessa. Jos vaatimusmäärittelytyössä epäonnistutaan syystä tai toisesta, ei uusi tietojärjestelmä todennäköisesti täytä asiakkaan tarpeita. Jo kehitysprojektin alussa tuleekin varmistaa, että vaatimusmäärittelyyn panostetaan riittävästi, jotta sillä on edellytykset onnistua. El Emamin ja Madhavjin [ELM95, 208-209] mukaan vaatimusmäärittelytyöhön käytettävät kustannukset suhteutettuna koko projektin kustannuksiin ovat eräs vaatimusmäärittelyn onnistumistekijä. Kustannusten suhde vaikuttaa olennaisesti vaatimusmäärittelyprosessin kannattavuuteen ja tukee sitä kautta vaatimusmäärittelyn onnistumista. Hofmann ja Lehner [HoL01, 65] tarkentavat, että vaatimusmäärittelyn osuus kehitysprojektin kokonaispanostuksesta tulisi olla noin 15 - 30 %. Ideaalitapauksessa vaatimusmäärittelytyöiden osuus olisi 28 %, josta vaatimusten kartutukseen varataan 11 %, mallinnukseen 10 % ja laadunvarmistukseen 7 %. Projektipäällikön kannattaa huomioida suositukset projektin suunnittelussa.

Aika- ja kustannuspanostuksen ohella vaatimusmäärittelyn onnistuminen edellyttää osaavia henkilöresursseja. Hofmannin ja Lehnerin [HoL01, 65] mukaan yksi vaatimusmäärittelytyön parhaista käytännöistä onkin osaavan ja kokeneen projektipäällikön sekä tiimin kiinnittäminen kehitysprojektiin. Käytännössä organisaatioissa on useita projekteja päällekkäin, eikä ole mahdollista kiinnittää parhaita osaajia niihin kaikkiin. Hofmann ja Lehner ehdottivat ratkaisuksi osa-aikaisten konkareiden käyttöä, jolloin syväosaajat osallistuvat projekteihin pienemmällä työpanoksella. Sähköisen terveydenhuollon tietojärjestelmät ovat kuitenkin turvallisuuskriittisiä, monimutkaisia järjestelmiä.

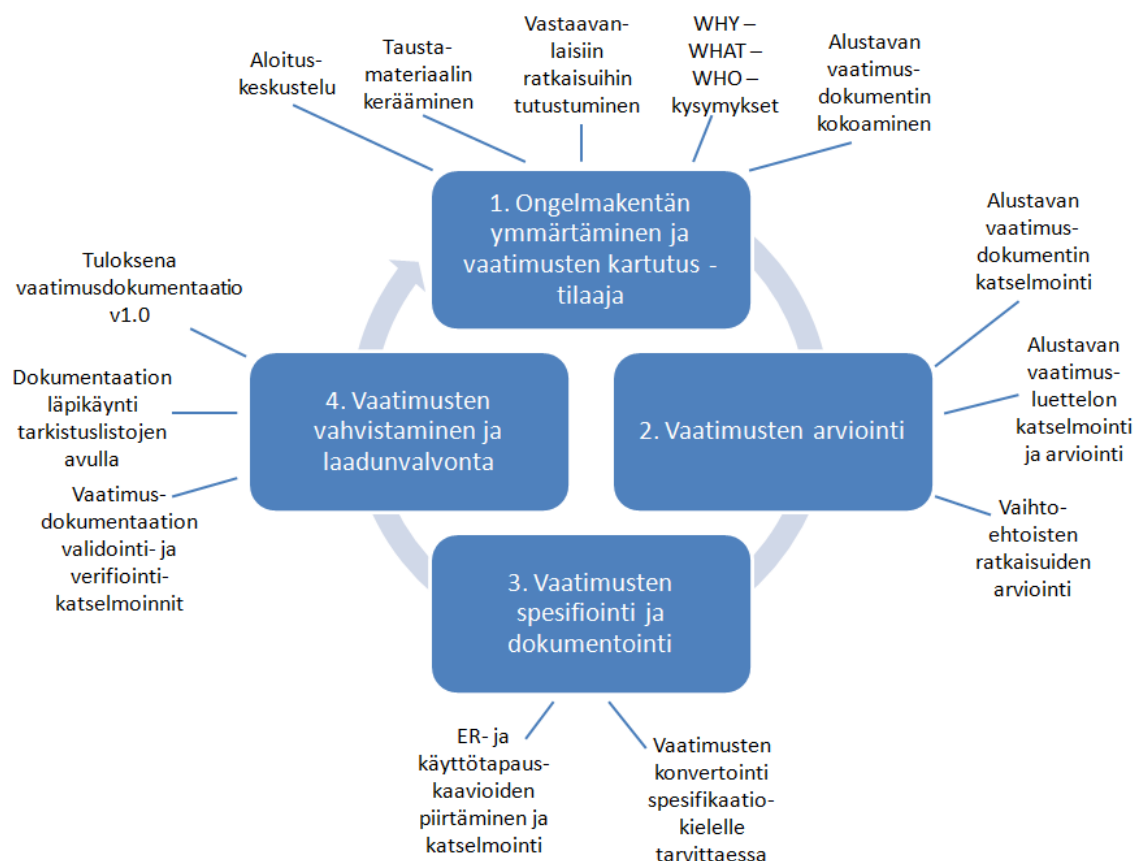
Tässä mallissa suositellaan, että tällaisten järjestelmien vaatimusmäärittelytyöhön kiinnitetään ainakin yksi vaatimusmäärittelytyön syväosaaja täyspäiväisesti. Hänen ohellaan voidaan toki hyödyntää vähemmän kokeneita henkilöitä suorittamaan tehtäviä hänen ohjauksessaan. Vaatimusmäärittelytyön syväosaajan ohella vaatimusmäärittelytyöhön (ainakin katselmointeihin) on tarpeen osallistua kehitystiimistä suunnittelijoita, toteuttajia ja testajia, jotta he voivat jo hyvissä ajoin orientoitua tuleviin tehtäviinsä. Lisäksi sähköisen terveydenhuollon tietojärjestelmien erityisluonteen vuoksi olisi tarpeen nimetä projektin käyttöön toimialan lakien ja määräysten asiantuntija sekä turvallisuuskriittisten järjestelmien asiantuntija. Kehitysprojektin projektipäällikön tulee siis huomioida edellä esitetyt asiat ja varata vaatimusmäärittelyyn riittävästi aikaa ja osaavia resursseja.

## **4.2 *Vaatimusmäärittelyn ensimmäinen iteraatiovariaatio***

Tutkielman teoriaosuudessa luvussa 2.2 esitettiin vaatimusmäärittelyprosessi, joka sisältää van Lamsweerden mukaan viisi työvaihetta [Lam09, 34-35]. Ne toistuvat iteratiivisesti kehitysprojektiin aikana tarkentaen asteittain tulevan järjestelmään kohdistuvia vaatimuksia työn edetessä. Kuvassa 9 esitellään prosessiin pohjautuvan mallin ensimmäinen iteraatiovariaatio ja sen aikana suoritettavat toimenpiteet. Ensimmäinen iteraatiovariaatio on suositeltavaa suorittaa heti kehitysprojektin alussa, sillä sen avulla saadaan esille koko projektia rajaavia tavoitteita, kuten aikataulu ja budjetti.

Kuten kuvasta 9 ilmenee, mallin ensimmäisen iteraatiovariaation ensimmäisessä työvaiheessa yhdistyy kaksi van Lamsweerden määrittelemää työvaihetta: ongelmakentän ymmärtäminen ja vaatimusten kartutus [Lam09, 30-32]. Vaiheen tavoitteena on hahmottaa kehitettävän järjestelmän avulla ratkaistava ongelma sekä siihen liittyvät reunaehdot ja oletukset. Lisäksi sen aikana pyritään tunnistamaan alustavia vaatimusehdokkaita ja näihin pohjautuvia vaihtoehtoisia ratkaisuja. Selvittelytyön tulokset kirjataan alustavaan vaatimusdokumenttiin, joka tulevien työvaiheiden myötä jalostuu varsinaiseksi vaatimusdokumentiksi. Mahdollisuuksien mukaan kannattaa hyödyntää organisaation omaa tai jotakin hyväksi todettua mallipohjaa esimerkiksi Internetistä, jotta työryhmä ei joudu käyttämään turhaa energiaa oman pohjan kehittelyyn. Hyväksytyn mallipohjan käyttö myös pienentää riskiä, että jotakin olennaista unohtuu kirjata, ja samalla määrämuotoinen rakenne parantaa dokumentaation laatua. Hofmannin ja Lehnerin [HoL01, 65] mu-

kaan valmiiden mallipohjien ja esimerkkien hyödyntäminen onkin yksi vaatimusmäärittelytyön parhaita käytäntöjä.



Kuva 9. Vaatimusmäärittelymallin ensimmäisen iteraatiovariaation toiminnot.

#### *Iteraatiovariaatio 1 - ongelmakentän ymmärtäminen ja vaatimusten kartutus*

Ensimmäisessä iteraatiovariaatiossa keskitytään erityisesti tilaajan näkökulmaan, jotta ongelmakentästä ja tavoitteista saadaan ymmärrystä aluksi yleisellä tasolla. Suomen sähköisen lääkemääräyksen tietojärjestelmäratkaisun tilaajana voidaan pitää sosiaali- ja terveysministeriön Sähköinen lääkemääräys -hankkeen johtoryhmää [Säh15a, 6]. Se määritteli sähköiselle lääkemääräykselle vision ja kiinnitti ylimmän tason tavoitteet. Mallin esittelyssä tilaaja käsitetään selkeyden vuoksi yhdeksi henkilöksi, vaikka sähköisen lääkemääräyksen tapauksessa tilaajan rooli käsitti useita henkilöitä.



Kuvan 9 mukaisesti ensimmäisen vaatimusmäärittelyiteraation alussa pidetään tilaajan kanssa aloituskeskustelu, joka voidaan tarvittaessa jakaa useampaan sessioon. Aloituskeskustelun tavoitteena on luoda yleiskäsitys siitä, millaisen ongelmakentän äärellä ollaan. Samalla motivoidaan ja sitoutetaan tilaajaa, sillä Hofmannin ja Lehnerin [HoL01, 65] mukaan vaatimusmäärittelyn onnistumisen kannalta on erittäin tärkeää, että hän osallistuu muiden sidosryhmäedustajien ohella työskentelyyn aktiivisesti koko vaatimusmäärittelyprosessin ajan.

Tilaajaa pyydetään esittelemään nykyistä ratkaisua ja siihen liittyviä toimintaprosesseja, eli miten lääkemääräyksiä tällä hetkellä kirjoitetaan ja toimitetaan. Mallintamalla paperisen lääkemääräyksen kirjoitus- ja toimitusprosessit saadaan käsitys siitä, kuinka asia on tällä hetkellä ratkaistu; esimerkiksi kuinka paperinen lääkemääräys syntyy, mihin asioihin siinä tulee ottaa kantaa, mitä toimijoita prosesseihin liittyy ja niin edelleen. Jos tilaajalla ei ole tarkkaa tietämystä toimintatavoista ja prosesseista, hän kuitenkin osannee nimetä alueen asiantuntijoita, jotka voivat esitellä nykyratkaisua. Jo tässä vaiheessa kannattaa kirjata ylös vaatimusehdokkaita, kun sellaisia tunnistetaan. Samalla aloitetaan sanaston kokoaminen, eli kaikki aihepiiriin liittyvät termit kirjataan ylös ja määritellään yksiselitteisesti. Tämä helpottaa yhteistyötä sidosryhmien kanssa ja on olennaista yhteisen ymmärryksen luomiseksi sekä väärinkäsitysten välttämiseksi.

Tässä esimerkissä vaatimukset suositellaan dokumentoitavaksi luonnollisella kielellä (puhekielellä). Tarvittaessa voidaan käyttää myös jotakin spesifikaatiokieltä, jos se katsotaan tarpeelliseksi ja jos kaikki osapuolet osaavat tulkita sitä oikein. Vaatimukset kirjataan alustavan vaatimusdokumentin liitteeksi sijoitettavaan vaatimusluetteloon. Kuten varsinainen vaatimusdokumentti, myös vaatimusluettelo kannattaa laatia määrämuotoiseen taulukkoon jonkin valmiin mallipohjan mukaisesti. Tämä helpottaa luettavuutta ja vaatimusten jatkokäsittelyä. Erityistä huomiota kannattaa heti dokumentoinnin alkuvaiheessa kiinnittää jäljitettävyyteen. Käytännössä tämä tarkoittaa sitä, että jokaisen vaatimuksen tai reunaehdon yhteyteen kirjataan sen esittäjätaho sekä mahdolliset suhteet ja riippuvuudet muihin vaatimuksiin tai elementteihin, kuten testitapauksiin. Hofmann ja Lehner [HoL01, 66] suosittavat jäljitettävyysematriisin laatimista yhtenä vaatimusmäärittelyn parhaista käytännöistä. Matriisin avulla on helposti todennettavissa esimerkiksi se, mihin kaikkialle muutos jossakin vaatimuksessa vaikuttaa.

Aloituskustelussa tilaajan kanssa käydään läpi, mitä lakeja, säädöksiä, määräyksiä ja standardeja tulevan järjestelmän tulee noudattaa. Koska terveydenhuolto ja siihen sisältyvät lääkemääräykset ovat tarkoin säännelty toimiala, kannattaa tutustua hyvin varhaisessa vaiheessa kaikkiin aihepiiriä sääteleviin lakeihin, säädöksiin ja määräyksiin. Niistä nousee olennaisia tavoitteita ja reunaehtoja, joita ei voida tulevassa tietojärjestelmäratkaisussa ohittaa tai soveltaa vapaasti. Esimerkiksi Sosiaali- ja terveysministeriön asetus lääkkeen määräämisestä [Sos10, 2. luku] määrittelee, kuka voi määrätä lääkettä. Tulevan tietojärjestelmän on mahdollistettava asetuksen noudattaminen kirjaimellisesti. Lisäksi laki sähköisestä lääkemääräyksestä [Lak14a, 19 pykälä] sanelee useita vaatimuksia ja reunaehtoja, joiden mukaisesti tulevan järjestelmän tulee toimia. Laissa esimerkiksi määritellään, että reseptit tallennetaan reseptiarkistoon, jossa niitä säilytetään 20 vuotta. Lainsäädännön monimutkaisuuden vuoksi olisi toivottavaa, että keskusteluun osallistuisi joku toimialan lakeja tunteja henkilö, kuten organisaation lakimies. Myös kaikki lakeihin ja määräyksiin pohjautuvat vaatimukset ja reunaehdot kirjataan tarkoin ylös.

Aloituskustelussa kartoitetaan myös mahdollisia kirjallisia vaatimuslähteitä ja muuta taustamateriaalia. Hofmann ja Lehner [Hol01, 65] korostavat, että vaatimusmäärittelytyössä on erittäin tärkeää tunnistaa ja hyödyntää kaikki mahdolliset vaatimuslähteet. Näin parannetaan vaatimusten kattavuutta, joka on suoraan verrannollinen tulevan järjestelmän toimivuuteen ja hyödyllisyyteen. Tilaajalta pyydetään tutustumista varten tarvittavaa taustamateriaalia, kuten liiketoimintasuunnitelmia, organisaatiokaavioita, nykyisen järjestelmän määrittely- ja suunnitteludokumentteja tai käyttöraportteja. Niistä saadaan lisäymmärrystä aihealueesta ja toimintaorganisaatiosta.

Aloituskustelun aikana saadaan siis karkea käsitys siitä, millaisesta ongelmakentästä on kyse. Sen jälkeen aletaan selvittää ongelmakentän tarkempia yksityiskohtia, kuten tavoitteita ja toimijoita. Niiden selvittämisessä käytetään apuna van Lamsweerden [Lam09, 12-16] esittelemää WHY-WHAT-WHO -kysymyspatteristoa. Kysymyspatteriston mukaisesti keskitytään ensin saamaan esille ydinongelma, eli kysytään uuden järjestelmän tilaajalta WHY-kysymys. Sen avulla pyritään selvittämään, miksi nykyratkaisu ei toimi, miksi tulevaa järjestelmää tarvitaan ja mihin halutaan muutosta. Sähköisen lääkemääräyshankkeen johtoryhmä tuo todennäköisesti esille, että paperiset lääkemääräykset ovat vanhanaikaisia, sillä niihin liittyy erilaisia riskitekijöitä aina väärin tulkit-

semisestä väärentämisen helppouteen. Lisäksi saatetaan nähdä puutteena se, että paperimuotoiset lääkemääräykset ja niiden toimitukset ovat sidoksissa asiakkaiden kotimaahan, ja tulevaisuudessa halutaan mahdollistaa maiden rajat ylittävä toiminta. Tässä vaiheessa WHY-kysymys voi toki nostaa esille muitakin ongelmakohtia, joihin tulevan ratkaisun toivotaan tuovan parannusta, ja ne kaikki kirjataan ylös.

Seuraavaksi aletaan selvittää tulevan järjestelmän tarkempia yksityiskohtia kysymällä tilaajalta WHAT. Kysymyksen avulla saadaan selville, mitä tulevan järjestelmän odotetaan tekevän, eli miten edellä havaitut ongelmat ratkaistaan. Samalla muodostuu alustava käsitys siitä, millaisia toiminnallisuuksia ja ominaisuuksia tulevan ratkaisun tulee voida tarjota käyttäjilleen. Tilaajalta saadaan tietoon todennäköisesti korkean tason tavoitteita ja toiminnallisuuksia, jotka tulevan järjestelmän tulee täyttää. Hän voi esimerkiksi vaatia, että tulevan järjestelmän avulla on voitava sekä kirjoittaa että toimittaa lääkemääräyksiä. Lisäksi hän voi tuoda esille, että uuden järjestelmän myötä lääkemääräysten tulee olla yksiselitteisesti tulkittavissa ja että niitä ei voi väärentää. Jälleen kaikki esille nousevat yksityiskohdat kirjataan ylös alustavaan vaatimusdokumenttiin.

On hyvä muistaa, että tilaaja katsoo kokonaisuutta usein hieman eri näkökulmasta kuin järjestelmän käyttäjät. Hän saattaa pitää tärkeimpänä asiana kustannussäästöjä tai tehokkuutta ja pohjaa vastauksensa tähän näkökulmaan. Tilaaja voi esimerkiksi haluta, että järjestelmän on tuotettava säästöjä mahdollistamalla lääkemääräyksen kirjoittamisen lyhyemmässä ajassa kuin mitä paperisen lääkemääräyksen kirjoittamiseen kuluu. Sen sijaan hän ei välttämättä osaa kuvailla nykyratkaisun tarkempia ongelmakohtia ja niistä johdettavia tulevan ratkaisun toivottavia ominaisuuksia yhtä hyvin kuin järjestelmän varsinaiset käyttäjät. Seuraavissa vaatimusmäärittelytyön iteraatioissa WHY-WHAT-WHO -kysymykset tullaankin esittämään myös nykyisen ja tulevan ratkaisun käyttäjille eli lääkäreille, apteekkareille ja näiden asiakkaille.

Nyt WHAT-kysymyksen avulla on saatu selville karkealla tasolla, mitä toiminnallisuuksia ja ominaisuuksia tilaaja odottaa tulevalta ratkaisulta. Seuraavaksi tulee vielä selvittää, mitä henkilöitä, tahoja, laitteita tai ohjelmistoja tulevaan ratkaisuun tulee sisällyttämään. Kysytään tilaajalta siis WHO-kysymys, eli ketkä tai mitkä tulevat vastaamaan ongelman ratkaisusta. Näin saadaan esille myös inhimilliset vaatimuslähteet eli sidoryhmät, joiden tulee osallistua vaatimusmäärittelyprosessiin. Sähköisen lääkemääräyk-

sen johtoryhmä voi esimerkiksi nimetä Suomen Lääkäriliiton, Suomen Apteekkariliiton, Fimean sekä apteekkien asiakkaat tahoiksi, jotka ovat osallisina tulevassa ratkaisussa. Lisäksi johtoryhmä voi listata tiedossaan olevat potilas- ja apteekkitietojärjestelmät sekä muut järjestelmät, laitteistot ja ohjelmistot, jotka liittyvät tulevaan ratkaisuun. Tilaaja ei välttämättä osaa nimetä kaikkia sidosryhmiä, tietojärjestelmiä tai ohjelmistoja, joita tulevaan ratkaisuun liittyy. Tätäkin osiota tarkennetaan myöhemmissä iteraatioissa haastatteleamalla tiedossa olevien sidosryhmien edustajia.

Seuraavaksi tilaajan esille nostamat sidosryhmät priorisoidaan. Priorisointikriteerinä on se, miten suuri merkitys kyseisellä sidosryhmällä on tulevalle järjestelmälle. Esimerkiksi apteekkariliitto voitaisiin priorisoida kaikkein merkityksellisimmäksi sidosryhmäksi, koska se on viime kädessä vastuussa lääkkeen luovuttamisesta asiakkaalle. Vastaavasti apteekin asiakkaat voitaisiin priorisoida kaikkein vähiten merkitykselliseksi sidosryhmäksi, koska he eivät tule itse suoraan käyttämään järjestelmää. Priorisointia voidaan myöhemmin käyttää apuna mahdollisten keskenään ristiriitaisten vaatimusten arvottamiseen.

Tilaajalta olisi hyvä saada myös tietoa siitä, mitä muita tähän kyseiseen asiakokonaisuuteen liittyviä projekteja tai hankkeita on meneillään tai alkamassa lähiaikoina. Sähköisen lääkemääräyksen tapauksessa oli jo varhain tiedossa, että sillä tulee olemaan liittymiä Kansallisen terveystietokannan (Kanta) kehittämishankkeeseen. Näin ollen olisi erittäin tärkeää, että kehitysprojektilla on toimivat yhteydet tällaisiin rinnakkaisprojekteihin ja -hankkeisiin, jotta tarvittavat liittymät ja muut yhteiset ratkaisut osataan huomioida ajoissa. Tunnistetut rinnakkaisprojektit kirjataan tässä vaiheessa sidosryhmiksi, joiden edustajat kutsutaan myöhemmin mukaan vaatimusmäärittelyyn.

Edellä mainittujen toimien avulla on saatu muodostettua tilaajan avulla alustava käsitys toimialasta ja ongelmakentästä sekä kerätty alustavaan vaatimusluetteloön joitakin vaatimusehdokkaita. Seuraavaksi tutustutaan muiden maiden vastaavantyyppisiin tietojärjestelmiin ja ratkaisuihin sekä mahdollisiin toimialalle suositeltuihin standardeihin. Suomi ei suinkaan ollut ensimmäinen maa, joka alkoi työstää lääkemääräyksiä sähköiseen muotoon. Vastaavia ratkaisuja on ollut kehitteillä muuallakin; esimerkkinä voidaan mainita vuosina 2008-2014 käynnissä ollut eurooppalainen epSOS-hanke [Mik14]. EpSOS-hankkeen tarkoituksena oli kokeilla potilastietojen, kuten sähköisten lääkemäärä-

ysten, toimittamista yli maarajojen. Yhteentoimivat ratkaisut ja standardit mahdollistavat tulevaisuudessa tällaisen laajemman, valtakunnan rajat ylittävän yhteistyön lääkemääräyksissä eri maiden välillä. Apuna kannattaa käyttää esimerkiksi luvussa 3.1 mainittuja IHE:n laatimia profiileita, jotka mahdollistavat tietyn osa-alueen järjestelmien integroinnin ja yhteentoimivuuden. Standardeista ja profiileista voidaan muodostaa eitoiminnallisia arkkitehtuurivaatimuksia. Lisäksi muualla jo käytössä olevien ratkaisuiden toimivuudesta on ehkä saatavilla kokemuksia ja arvioita, jotka auttavat hahmottamaan ja rajaamaan erilaisia ratkaisu- ja teknologiavaihtoehtoja tulevalle järjestelmälle. Esimerkkien käyttö onkin eräs vaatimusmäärittelyn parhaita käytäntöjä [HoL01, 65].

Ensimmäisen työvaiheen tuloksena on sähköisen lääkemääräystietojärjestelmän alustava vaatimusdokumentti. Se sisältää kuvausta seuraavista asioista: toimiala ja ongelmakenttä lääkemääräysten ja toimitusten näkökulmasta, olemassa oleva järjestelmä eli paperiset lääkemääräykset ja niiden toimintaprosessit, toimintaorganisaatio ja tunnistetut sidosryhmät, tunnistetut ongelmat ja mahdollisuudet sekä tähän mennessä tiedossa olevat ratkaisuvaihtoehdot. Sen liitteeksi vaatimusluetteloon on kirjattu haastatteluissa ja taustatutkimuksissa tunnistettuja vaatimusehdokkaita, jotka voivat vielä tässä vaiheessa olla hyvin epämääräisiä tai jopa keskenään ristiriitaisia. Lisäksi sen liitteenä on sanasto, josta ilmenee aihealueeseen liittyvä termistö.

#### *Iteraatiovariaatio 1 - vaatimusten arviointi*

Seuraavaksi iteraatiovariaatiossa siirrytään van Lamsweerden vaatimusmäärittelyprosessin seuraavaan vaiheeseen, joka on vaatimusten arviointi [Lam09, 32-33]. Arviointivaiheessa alustava vaatimusdokumentti liitteineen katselmoidaan yhdessä tilaajan kanssa. Hofmann ja Lehner [HoL01, 66] tuovat esille, että katselmoinnit ja läpikäynnit ovat merkittävässä roolissa vaatimusmäärittelytyön onnistumisessa. Niiden avulla voidaan varmistaa, että vaatimukset on kirjattu oikein ja ymmärrettävästi. Lisäksi katselmoinnit sitouttavat tilaajaa ja sidosryhmäedustajia, ja säännöllinen vuorovaikutus mahdollistaa palautteen antamisen ja saamisen. Avoin ilmapiiri ja hyvät suhteet sidosryhmiin ovatkin keskeisessä roolissa vaatimusmäärittelyn onnistumisessa [HoL01, 65]. Niiden myötä saadaan esille eri näkökulmia ja voidaan välttää väärinkäsityksiä, jotka voisivat johtaa viivästyksiin tai toimimattomiin järjestelmiin.

Katselmoinnin tavoitteena on varmistua kirjausten oikeellisuudesta. Mahdolliset asia-  
virheet korjataan heti katselmoinnin yhteydessä. Lisäksi osallistujien kesken pyritään  
saavuttamaan yhteinen näkemys siihen mennessä tunnistetuista vaatimuksista. Koska  
vaatimusmäärittelyyn on tässä vaiheessa osallistunut vain tilaajan edustajia, ei suuria  
näkemyseroja tai ristiriitoja pitäisi ilmetä. Mahdollisista näkemyseroista neuvotellaan ja  
vaatimuksia muokataan niin kauan, että ne tyydyttävät kaikkia osapuolia ja ovat yksise-  
litteisiä. Lopuksi vaatimusluetteloon kirjatut vaatimukset priorisoidaan tärkeysjärjestyk-  
seen. Hofmannin ja Lehnerin [HoL01, 65-66] mukaan priorisointi on edellytys vaati-  
musmäärittelyn onnistumiselle, sillä sen avulla projektissa pystytään kohdentamaan  
resurssit tilaajan näkökulmasta olennaisimpien vaatimusten työstämiseen.

Samalla arvioidaan myös dokumenttiin kirjattuja vaihtoehtoisia ratkaisuja. Niihin liitty-  
vät kustannusarviot, mahdollisuudet ja riskit, kuten turvallisuuteen tai kehitystyön riskit  
kirjataan alustavaan vaatimusdokumenttiin. Lopuksi vaihtoehtoiset ratkaisut priorisoi-  
daan panos-tuotos-riski -näkökulmien pohjalta, jotta löydetään kaikkein kustannuste-  
hokkain, toimivin ja pieniriskisin vaihtoehto. Tämä mahdollistaa resurssien järkevän  
kohdentamisen, mutta jättää option jonkin toisen vaihtoehdon valitsemiselle myöhem-  
min, mikäli se koetaan järkeväksi. Sähköisen lääkemääräysjärjestelmän kohdalla vaih-  
toehtoisia ratkaisuita voisivat olla esimerkiksi itsenäinen mammuttijärjestelmä, joka  
kattaa myös nykyiset potilas- ja apteekkitietojärjestelmät, ja olemassa olevia potilas- ja  
apteekkitietojärjestelmiä hyödyntävä ja täydentävä, integroitava järjestelmä. Molempien  
kannalta arvioidaan siis toteutuskustannuksia, ratkaisuiden tarjoamia mahdollisuuksia  
sekä niihin liittyviä riskejä. Vaiheen lopputuloksena on päivitetty versio alustavasta vaa-  
timusdokumentista, josta ilmenee tässä vaiheessa kiinnitetyt ja priorisoidut yhteiset vaa-  
timukset sekä tulevaan järjestelmään liittyvät oletukset.

#### *Iteraatiovariaatio 1 - vaatimusten spesifointi ja dokumentointi*

Kun vaatimukset on arvioitu, siirrytään iteraatiovariaatioissa van Lamsweerden näke-  
myksen mukaisesti seuraavaan vaiheeseen eli vaatimusten spesifointiin ja dokumen-  
tointiin [Lam09, 33]. Vaiheen yleistavoitteena on edellisessä vaiheessa muotoiltujen  
vaatimusten ja niihin liittyvien järjestelmäominaisuuksien dokumentointi. Tässä esimer-  
kissä vaatimuksia tarkennettiin hyväksyttävään muotoonsa ja ne kirjattiin alustavan vaa-  
timusdokumentin liitteeksi vaatimusluetteloon jo edellisessä vaiheessa. Vaatimusluette-

loa on hyvä hioa vielä tässä vaiheessa esimerkiksi jaotteleamalla tunnistetut ei-toiminnalliset vaatimukset kuvan 5 mukaisiin luokkiin. Näin saadaan käsitystä siitä, mihin luokkiin on jo tunnistettu vaatimuksia ja mitkä luokat ovat vielä mahdollisesti käsittelemättä. Lisäksi, jos kehitysprojektissa olisi jostain syystä päädytty dokumentoimaan vaatimukset jollakin formaalilla spesifikaatiokielellä, voitaisiin yhteiset vaatimukset konvertoida tässä vaiheessa sen mukaisiksi.

Tämän jälkeen mallinnetaan vaatimuksia ja tulevaa järjestelmää piirtämällä selkiyttäviä kaavioita. Ensimmäisessä iteraatiossa kannattaa aloittaa sellaisista kaavioista, joihin liittyviä tietoja on käytettävissä tässä vaiheessa. Tällaisia kaavioita ovat esimerkiksi ongelmakentän käsitteitä ja tietosisältöjä kuvaavat ER-kaaviot sekä järjestelmän tiedoille suoritettavia operaatioita kuvaavat käyttötapauskaaviot. Kaavioita voidaan piirtää sekä nykyjärjestelmästä että tulevasta järjestelmästä. Näin saadaan dokumentoitua nykytilanne ja samalla alustettua pohdintaa siitä, millaisia muutoksia tai päivityksiä tulevaan järjestelmään on tarpeen tehdä. On hyvä muistaa, ettei kaavioita ole tarkoitus saada täydellisiksi tässä vaiheessa, vaan ne kuvataan tämänhetkisten tietojen pohjalta, ja niitä tarkennetaan myöhemmissä iteraatioissa. Laaditut kaaviot käydään vielä läpi yhdessä tilaajan kanssa ja varmistetaan, että niihin on kuvattu asiat oikein. Lopuksi kaaviot liitetään vaatimusdokumentin liitteiksi. Vaiheen lopputuloksena on siis vaatimusdokumentti ja sen liitteisiin dokumentoidut vaatimukset.

#### *Iteraatiovariaatio 1 - vaatimusten vahvistaminen ja laadunvalvonta*

Seuraavaksi siirrytään ensimmäisen iteraatiovariaation viimeiseen työvaiheeseen, joka on vaatimusten vahvistaminen ja laadunvalvonta. Van Lamsweerden [Lam09, 33, 187-188] mukaan vaiheen tavoitteena on varmistaa, että dokumentoidut vaatimukset täyttävät sidosryhmien (tässä ensimmäisessä iteraatiovariaatiossa tilaajan) todelliset tarpeet eli ovat valideja. Varsinainen työskentely tapahtuu katselmoinnin tai läpikäynnin muodossa yhdessä tilaajan kanssa. Samalla vaatimukset verifioidaan eli varmistetaan, ettei vaatimusmäärittelydokumentaatioissa ole keskenään ristiriitaisia vaatimuksia. Vaatimusten tulee myös olla linjassa dokumentoitujen oletusten ja rajoitteiden kanssa.

Laadunvalvonnan tukena voidaan tarvittaessa käyttää apuna erilaisia tarkistuslistoja. Van Lamsweerden [Lam09, 191-192] koostaman tarkistuslistan avulla dokumentaatiosta voidaan etsiä esimerkiksi ristiriitoihin, soveltuvuuteen, mitattavuuteen ja ymmärrettä-

vyytteen liittyviä virheitä. Lisäksi vaatimusdokumentaation rakennetta tarkastellaan tässä iteraatiovariaatiossa kahden van Lamsweerden [Lam09, 35-36] määrittelemän laatutavoitteen näkökulmasta. Vaatimusdokumentaation osalta varmistetaan, että se täyttää *muunneltavuus*-tavoitteen. Sen mukaisesti dokumentaation tulee olla koostettu niin, että sitä on helppo laajentaa ja työstää tulevilla iteraatioilla. Samalla tulee varmistua siitä, että dokumentaation rakenne on looginen ja yhtenäinen ja että sen elementtien riippuvuudet ja syy-seuraus-suhteet on kuvattu selkeästi. Näin dokumentaatio täyttää myös *hyvän rakenteen* laatutavoitteen. Näiden tavoitteiden saavuttamista helpottaa hyvien mallipohjien käyttö, kuten jo aiemmin todettiin. Havaitut ristiriidat ja virheet korjataan, ja ensimmäisen iteraatiovariaation viimeisen vaiheen lopputuloksena on tilaajan näkökulmasta tunnistetut ja vahvistetut vaatimukset sisältävä vaatimusdokumentti liitteineen. Tässä vaiheessa dokumentaatio kirjataan versioksi 1.0, joka osoittaa, että kyseessä on ensimmäisessä iteraatiovariaatiossa katselmoitu ja vahvistettu dokumentaatio.

### 4.3 *Vaatimusmäärittelyn toinen iteraatiovariaatio*

Ensimmäisessä iteraatiovariaatiossa pääpaino oli toimialan ja ongelmakentän ymmärtämisessä erityisesti tilaajan näkökulmasta. Sen aikana saatiin koottua myös joitakin tilaajan vaatimuksia tulevalle järjestelmälle ja kartoitettiin alustavia ratkaisuvaihtoehtoja. Seuraavaksi toisessa iteraatiovariaatiossa lähdetään tarkastelemaan kokonaisuutta tilaajalle esitetyn WHO-kysymyksen avulla tunnistettujen sidosryhmien näkökulmasta eli esimerkissämme Suomen Lääkäriliiton, Suomen Apteekkariliiton sekä Fimean edustajilta. Tavoitteena on saada mahdollisimman kattava käsitys ongelmakentästä.

Olisi toivottavaa, että jokaisesta sidosryhmästä osallistuisi vaatimusmäärittelytyöhön ainakin kaksi edustajaa. Tällöin edustajat saavat toisistaan tukea ja vahvistusta juuri oman sidosryhmänsä tarpeiden kuvaamiseen ja konkretisointiin. Lisäksi he voivat sijaita toisiaan esimerkiksi sairastapauksissa, jolloin työtä voidaan edistää, vaikka toinen edustaja olisikin jostain syystä estynyt osallistumasta johonkin kokoukseen. Myös tilaajan edustajan on hyödyllistä osallistua sidosryhmien näkemyksiin ja tarpeisiin keskittyviin kokouksiin ja läpikäynteihin. Näin hän saa tietoa siitä, millaisia konkreettisia tarpeita tulevaan järjestelmään kohdistuu esimerkiksi käyttäjien taholta ja voi tehdä järkeviä päätöksiä budjetin ja aikataulun suhteen. Toinen iteraatiovariaatio kannattaa ajoittaa heti ensimmäisen iteraatiovariaation perään kehitysprojektin alkuun, jotta tulevan järjes-



telmän vaatimukset saadaan selville riittävän kattavasti tulevia vaiheita varten. Vaatimusmäärittelymallin toinen iteraatiovariaatio ja sen aikana suoritettavat toimenpiteet esitellään kuvassa 10.



Kuva 10. Vaatimusmäärittelymallin toisen iteraatiovariaation toiminnot.

#### *Iteraatiovariaatio 2 - ongelmakentän ymmärtäminen ja vaatimusten kartutus*

Toisen iteraatiovariaation ensimmäisessä työvaiheessa pyritään ymmärtämään ongelma- kenttää entistä syvällisemmin eri näkökulmista. Samalla kartutetaan vaatimuksia tulevalle järjestelmälle, nyt sidosryhmien ja käyttäjien kannalta. Kuten kuvasta 10 ilmenee, iteraatiovariaation aluksi tähän mennessä tunnistettujen sidosryhmien edustajat kutsutaan koolle vaatimusmäärittelytyön aloituskokoukseen. Aloituskokouksen tarkoituksena on antaa kaikille osallistujille yhteinen käsitys siitä, mistä kyseisessä kehitysprojektissa on kyse, eli millaista järjestelmää ollaan kehittämässä ja miksi.

Koska läsnäolijat ovat todennäköisesti terveydenhuollon ja lääketeollisuuden ammattilaisia, heillä ei välttämättä ole tietoa tai kokemusta tietojärjestelmien kehitystyöstä tai siihen sisältyvästä vaatimusmäärittelystä. Tämän vuoksi on myös tärkeää esitellä vaati-

musmäärittelytyö sekä siihen kuuluvat työvaiheet ja käytänteet. Näin osallistujat saavat käsityksen siitä, millaisesta työkokonaisuudesta vaatimusmäärittelyssä on kyse ja mikä vaikutus sillä on tulevaan järjestelmään. Samalla käydään läpi työskentelytavat ja sitoutetaan osallistujat niihin ja koko vaatimusmäärittelytyöhön. Kuten jo aiemmin todettiin, Hofmannin ja Lehnerin [HoL01, 65] mukaan on vaatimusmäärittelyn onnistumisen kannalta erittäin tärkeää, että tulevan järjestelmän tilaajat, käyttäjät ja muut sidosryhmäedustajat osallistuvat vaatimusmäärittelyprosessiin sen alusta loppuun saakka. Lopuksi sidosryhmäedustajille esitellään työstetyt dokumentit, jotta he saavat ymmärryksen pait-si esille saaduista seikoista, myös tuotettavan dokumentaation esitysmuodosta.

Aloituskokouksen jälkeen työskentely jatkuu työpajatyöskentelynä soveltuvissa pienryhmissä. Työskentely pienemmissä ryhmissä on hyödyllistä siksi, että sähköisen terveydenhuollon tietojärjestelmien käyttäjäkunta voi olla hyvinkin heterogeenistä. Isossa ryhmässä jotakin olennaista voisi jäädä huomaamatta. Työpajatyöskentely aloitetaan nykyjärjestelmän analysoinnilla ja nykyisten toimintaprosessien mallintamisella eri toimijoiden näkökulmasta. Näin saadaan ymmärrystä siitä, miten nykyinen ratkaisu toimii. El Emam ja Madhavji korostavat, että nykyisen tietojärjestelmän analysoinnin kattavuudella on merkittävä vaikutus tulevan ratkaisun vaatimusmäärittelyn onnistumiselle [ELM95, 209], joten siihen kannattaa panostaa. Esimerkiksi apteekkeissa lääketoimitusten parissa työskenteleviä sidosryhmäedustajia pyydetään kertomaan ja esittelemään vaihe vaiheelta, kuinka he tällä hetkellä suorittavat lääketoimituksia. Näin saadaan myös selville, millaisia tietoja nykyisissä järjestelmissä käsitellään. Tietoja voidaan hyödyntää tulevan järjestelmän tietorakenteiden ja -suojausten suunnittelussa.

Työpajoissa saadun informaation pohjalta täydennetään edellisessä iteraatiovariaatioissa aloitettuja prosessikuvauksia ja mallinnetaan tulevaan järjestelmään liittyviä toimintaprosesseja. Oletettavasti esille nousisi liitteen 2 mukaisia asioita, eli toiminnallisuuksia, joita tulevan järjestelmän tulee tarjota käyttäjilleen. Lääkäriedustajat saattaisivat kertoa, että heidän pitää voida tarvittaessa korjata lääkemääräystä, jos siinä on jokin virhe. Vastaavasti farmaseuttiedustajat saattaisivat tuoda esille, että asiakkaiden kannalta voisi olla hyödyllistä, jos he pystyisivät kirjaamaan lääkemääräyksen uusimispyynnön.

Tunnistettuihin prosesseihin mallinnetaan myös niihin liittyvät skenaariot. Skenaarioissa kuvataan tulevan ratkaisun vuorovaikutusta ympäristönsä kanssa, ja ne auttavat hahmot-

tamaan tulevan järjestelmän toimintaa eri tilanteissa. Olennaista on saada selville niin sanotun normaaliskenaarion (eli miten esimerkiksi lääketoimitus tapahtuu normaalitilanteessa) ohella myös kaikki vaihtoehtoiset skenaariot eli poikkeustilanteet, joita joskus ilmenee syystä tai toisesta. Poikkeustilanne voisi olla esimerkiksi se, että määrätty lääke onkin loppu tai se, että asiakas ei pystykään maksamaan lääkkeitään kassalla. Mallintamalla tunnistettuihin prosesseihin liittyvät skenaariot saadaan ymmärrystä siitä, millaisista tilanteista myös tulevan järjestelmän pitäisi selvitä.

Mallinnuksen avulla saadaan mahdollisesti selville myös vaatimusehdokkaita, jotka kirjataan ylös edellisessä iteraatiovariaatiossa aloitettuun vaatimusluetteloon. Uudet vaatimusehdokkaat tulee dokumentoida niin, että niistä ilmenee vaatimuksen esittäjätahto ja esittämisajankohta. Tämä helpottaa vaatimusten arviointia ja tukee vaatimusten jäljitettävyyttä. Tilaajan edustajan on erittäin suositeltavaa osallistua pienryhmien toimintaan, jotta hän saa ymmärrystä eri sidosryhmien tarpeista ja näkökulmista myöhempiä päätöksentekoa varten. Lisäksi hän voi oivaltaa uusia tarpeita tai vaatimuksia tulevalle järjestelmälle.

Työpajoissa käydään myös läpi van Lamsweerden [Lam09, 12-16] WHY - WHAT - WHO -kysymyspatteristoa samalla periaatteella kuin ensimmäisessäkin iteraatiovariaatiossa. WHY-kysymyksen avulla pyritään selvittämään, mikä on ydinongelma kunkin sidosryhmän kannalta. WHAT-kysymys paljastaa, millaisia ominaisuuksia sidosryhmät toivovat tulevalta järjestelmältä. Lopuksi WHO-kysymyksen avulla saadaan tietoon mahdollisia uusia sidosryhmiä sekä tietojärjestelmiä, ohjelmistoja ja laitteistoja. Uudet sidosryhmät lisätään aiemmin tunnistettuihin sidosryhmiin ja priorisoidaan merkityksellisyyden perusteella. Uusien sidosryhmien edustajat myös kutsutaan mukaan seuraaviin vaatimusmäärittelyiteraatioihin. Ensimmäisessä iteraatiovariaatiossa koottua vaatimusdokumenttia ja sen liitteitä täydennetään ja päivitetään uusien havaintojen mukaan. Tärkeää on edelleen huolehtia dokumenttien versioinnista, jotta niistä selviää yksiselitteisesti, miten dokumenttia on edellisen vahvistusvaiheen jälkeen muokattu ja miksi. Mahdolliset ristiriidat korostetaan, jotta niihin muistetaan palata arviointivaiheessa.

Toisessa iteraatiovariaatiossa varataan vähintään yksi työpajakokoontuminen ei-toiminnallisten vaatimusten kartuttamiseen, jotta niihin tulee kiinnitettyä tarpeeksi huomiota. Ei-toiminnallisten vaatimusten avulla saadaan selville tarkempia yksityiskoh-

tia siitä, miten tulevan järjestelmän toivotaan toimivan ja tarjoavan palveluita käyttäjilleen. Työskentelyn tukena on suositeltavaa käyttää esimerkiksi kuvan 5 mukaista jaottelea, jotta eri näkökulmat muistetaan huomioida. Pysähtymällä pohtimaan esimerkiksi kuvan 5 alakategoriaa *turvallisuus sisäisen toiminnan kannalta* tai *suojaus ulkoisilta tekijöiltä* voidaan varmistaa, että myös tietosuojaan liittyvät ei-toiminnalliset vaatimukset tulee huomioida. Vastaavasti turvallisuuskriittisten järjestelmien kohdalla esimerkiksi luotettavuus- ja suorituskyykyvaatimukset on syytä kartoittaa erityisen tarkasti.

On otettava huomioon, etteivät sidosryhmien edustajat välttämättä osaa spontaanisti tuottaa ei-toiminnallisia vaatimuksia esimerkiksi suorituskyyvyn näkökulmasta. Tällöin heitä voidaan pyytää asettamaan vaatimuksia vaikkapa vasteaikaan kysymällä, kuinka nopeasti tulevan järjestelmän tulee hakea näytölle potilaan tiedot. Tilaaajan edustajalta voidaan saada kehitystyötä ohjaavia ei-toiminnallisia vaatimuksia kuten kustannus- tai tavoiteaikarajoitteita. Lisäksi tulevan järjestelmän tulevat ylläpitäjät voivat esittää esimerkiksi asennettavuus- tai ylläpidettävyyksivaatimuksia. Kaikki esille nousevat vaatimukset kirjataan vaatimusluetteloon ei-toiminnallisiin vaatimuksiin.

Jos työpajoihin ei saada riittävästi osallistujia, voi vaatimusmäärittelyn kattavuus vaarantua. Tällöin työpajoille vaihtoehtoisena työskentelytapana voidaan käyttää eri sidos- ja käyttäjäryhmille suunnattuja kirjallisia kyselyitä. Kyselyiden avulla selvitetään, mikä nykyratkaisussa toimii ja mikä ei, ja millaisia ominaisuuksia ja toiminnallisuuksia tulevalta järjestelmältä toivotaan. On kuitenkin muistettava, että kyselyt ovat kovin yksisuuntainen tiedonhankintatapa, ja väärinymmärrykset ovat aina mahdollisia puolin ja toisin. Kyselyiden ohella olisikin hyvä järjestää vielä tarkentavia haastatteluita, havainnointiin perustuvia esittelytilaisuuksia tai työpajoja, joissa käyttäjät voivat esitellä työtapaansa ja ajatuksiaan. Näin voidaan varmistua siitä, että sidosryhmien tarpeet on ymmärretty oikein ja saadaan selville konkreettisin esimerkein, millaisia toiminnallisuuksia ja ominaisuuksia esimerkiksi lääkärit, potilaat tai apteekin henkilökunta toivovat tulevalta järjestelmältä.

#### *Iteraatiovariaatio 2 - vaatimusten arviointi*

Seuraavaksi on vuorossa vaatimusten arviointivaihe. Toisessa iteraatiovariaatiossa se koostuu kahdesta katselmointikierrroksesta; pienryhmien katselmoinnista sekä yhteiskatselmoinnista. Ensin yleistä vaatimusdokumenttia katselmoidaan pienryhmissä ja var-

mistutaan siitä, että siihen kyseisen ryhmän osalta tehdyt täydennykset kuten prosessikuvaukset ovat oikeellisia. Tämän jälkeen pienryhmissä katselmoidaan vaatimusluettelosta kyseisen ryhmän tunnistamat uudet vaatimusehdokkaat. Katselmoinnin tarkoituksena on varmistaa, että vaatimusehdokkaat on kirjattu oikeellisesti ja yksiselitteisesti ja että pienryhmien edustajat ovat niistä yksimielisiä. Tarvittaessa käydään tarkentavia neuvotteluita ja muokataan ehdokkaita, jotta pienryhmien sisällä vallitsee yhteisymmärrys vaatimuksista.

Seuraavaksi vaatimusdokumentti ja -luettelo katselmoidaan yhdessä kaikkien sidosryhmäedustajien kanssa. Vaatimusdokumentin osalta varmistutaan siitä, että sen kirjaukset tyydyttävät kaikkia osapuolia ja ettei siinä ole ristiriitoja. Tämän jälkeen on vuorossa vaatimusluettelon yhteiskatselmointi. Koska vaatimuksia on kartutettu toisistaan erillisissä pienryhmissä, osa kirjatusta vaatimuksista voi olla päällekkäisiä tai ristiriitaisia. Tavoitteena onkin yhdistää päällekkäiset vaatimukset ja poistaa mahdolliset ristiriidat vaatimusehdokkaiden välillä. Neuvotteluita käydään siis niin kauan, että saadaan selville keskenään ristiriidattomat, yhteiset vaatimukset, joihin kaikki osapuolet voivat sitoutua. Lopuksi vaatimukset priorisoidaan tärkeysjärjestykseen. Priorisoinnissa tulee ehdottomasti olla mukana myös tilaajan edustaja, joka voi tarvittaessa ottaa kantaa vaatimusten prioriteetteihin, jos asiassa ei muuten päästä yhteisymmärrykseen.

Seuraavaksi arvioidaan vaihtoehtoisia ratkaisuja. Sidosryhmien esille tuomien yksityiskohtien valossa pysähdytään pohtimaan, onko ensimmäisessä iteraatiovariaatiossa tunnistettujen ratkaisujen lisäksi ilmennyt uusia vaihtoehtoja tulevalle ratkaisulle. Uudet ratkaisuvaihtoehdot kirjataan vaatimusdokumenttiin panos-tuotos-riski -analyysineen. Lisäksi esille on voinut nousta jo edellisessä iteraatiossa tunnistettuihin vaihtoehtoihin ratkaisuihin uusia näkökulmia kuten uusia riskejä. Vaihtoehtoiset ratkaisut priorisoidaan tarvittaessa uudelleen panos-tuotos-riski -näkökulmien pohjalta tavoitteena löytää kaikin kustannustehokkain, toimivin ja pieniriskisin vaihtoehto eli niin sanottu suositeltu ratkaisu.

El Emam ja Madhavji [EIM95, 208-209] korostavat, että käyttäjien yhteisymmärrys suositellusta ratkaisusta on yksi vaatimusmäärittelytyön onnistumistekijöistä. Yhteisymmärrys vaikuttaa myönteisesti käyttäjien tyytyväisyyteen ja sitoutumiseen, joka puolestaan on vaatimusmäärittelypalvelun laadun osatekijä ja vaikuttaa merkittävästi vaati-

musmäärittelyn onnistumiseen. Niinpä neuvotteluita käydään siihen saakka, että yhteisymmärrys löytyy. Tarvittaessa hyödynnetään aiemmin tehtyä sidosryhmäpriorisointia, jolloin merkittävimmiksi määritellyillä sidosryhmillä on enemmän sananvaltaa ratkaisun valinnassa.

Tilaajan näkökulmasta tulee myös varmistaa, että tilaajaorganisaatiolla on oikeasti kyky tehdä tarvittavat muutokset, jotta suositeltu ratkaisuvaihtoehto voidaan ottaa käyttöön [EIM95, 208-209]. Käytännössä tämä voi tarkoittaa esimerkiksi sitä, että tilaaja on valmis muuttamaan nykyistä organisaatiota, sen toimintaprosesseja tai infrastruktuuria tarvittavilta osin. Jos ratkaisun edellyttämiin muutoksiin ei pystytä sitoutumaan, kannattaa valita jokin toinen ratkaisuvaihtoehto, joka voitaisiin ottaa käyttöön pienemmillä muutoksilla. Sähköisen lääkemääräyksen tapauksessa on tässä kohden varmasti punnittu, olisiko järkevää tai edes mahdollista tavoitella yhtä suurta mammuttijärjestelmää integraation sijasta. Oletettavasti edellytyksiä yhtenäiselle ratkaisulle ei ollut, joten oli päädytty integraatioon, jossa hyödynnetään jo olemassa olevia potilas- ja apteekkitietojärjestelmiä (vertaa kuva 8). Samalla kannattaa varmistaa, että suositeltu ratkaisu on yhteensopiva organisaation strategian kanssa. El Emamin ja Madhavjin [EIM95, 208-209] mukaan organisaation kyky tehdä tarvittavat muutokset suositellun ratkaisun käyttöönottamiseksi sekä suositellun ratkaisun yhteensopivuus organisaation strategian kanssa tukevat kiinteästi suositellun ratkaisun sopivuutta organisaatioon. Se puolestaan vaikuttaa vaatimusmäärittelypalvelun laatuun ja siten tukee vaatimusmäärittelyn onnistumista.

Vaatimusdokumentin ja sen liitteiden kaksipuolainen katselmointi voidaan kokea työlääksi tai aikaa vieväksi. Se on kuitenkin perusteltua siksi, että sähköisen terveydenhuollon tietojärjestelmiin liittyy usein hyvin erilaisia sidosryhmiä omine tarpeineen. Näiden tarpeiden esille saanti ja dokumentointi on edellytys toimivalle tietojärjestelmälle. Samalla kaikkien vaatimusmäärittelyyn osallistuvien henkilöiden tulisi myös ymmärtää niin kokonaistilanne kuin eri sidosryhmienkin näkökulmat sekä se, mihin tulevassa ratkaisussa pyritään ja miksi. Näin kenellekään ei tule yllätyksenä, jos uusi järjestelmä toimiikin eri tavalla kuin itse oli alun perin ajatellut. El Emam ja Madhavji [EIM95, 208-209] korostavat, että käyttäjien ymmärrys siitä, mitä tuleva järjestelmä tekee ja ei tee, vaikuttaa suuresti käyttäjien tyytyväisyyteen ja sitoutumiseen. Heidän tutkimustensa mukaan tällä puolestaan on merkittävin vaikutus vaatimusmäärittelypalvelun laatuun ja sitä kautta koko vaatimusmäärittelyn onnistumiseen. Vaiheen lopputuloksena on päi-

vitetty versio vaatimusdokumentista liitteineen. Niistä ilmenevät tässä vaiheessa kiinnitetyt ja priorisoidut yhteiset vaatimukset sekä tulevaan järjestelmään liittyvät oletukset.

#### *Iteraatiovariaatio 2 - vaatimusten spesifointi ja dokumentointi*

Kun on saatu selville yhteiset vaatimukset, ne tulee vielä spesifioida ja dokumentoida. Työvaihe suoritetaan samalla periaatteella kuin ensimmäisessä iteraatiovariaatiossa. Vaatimusluetteloon kirjattuja vaatimuksia luokitellaan tarpeen mukaan sekä dokumentoidaan niihin liittyvää metatietoa, kuten keskinäisiä riippuvuuksia. Tarvittaessa tehdään konvertointi halutulle spesifikaatiokielelle, jos sellaista on päädytty käyttämään. Lisäksi täydennetään edellisessä iteraatiossa hahmoteltuja ER- ja käyttötapauskaavioita uuden tietämyksen pohjalta. Niiden ohella voidaan luonnostella myös tietovuo-, aktiviteetti-, viestiyhteys- tai tilakaavioita, jos ne koetaan tarpeellisiksi teknisten suunnitelmien tai kehitystyön tueksi. Kaaviot katselmoidaan erityisesti projektin teknisten suunnittelijoiden ja toteuttajien toimesta ja lisätään vaatimusdokumentin liitteiksi.

#### *Iteraatiovariaatio 2 - vaatimusten vahvistaminen ja laadunvalvonta*

Seuraavaksi on vuorossa toisen iteraatiovariaation viimeinen työvaihe, joka on vaatimusten vahvistaminen ja laadunvalvonta. Tässä iteraatiovariaatiossa vaiheeseen sisällytetään dokumentaation laajempi tarkastelu van Lamsweerden [Lam09, 35-36] määrittelemien vaatimusdokumenttien laatutavoitteiden näkökulmasta. Tuotetut dokumentit katselmoidaan ja varmistetaan, että kirjatut vaatimukset täyttävät sidosryhmien todelliset tarpeet eli ovat valideja. Tarvittaessa vaatimuksia täydennetään tai muokataan. Jos vaatimusten todetaan olevan valideja ja esitetyt ohjelmistovaatimukset ovat linjassa järjestelmävaatimusten kanssa, voidaan vaatimusmäärittelyn katsoa täyttävän *sopivuuden* laatutavoitteen. Vaatimusten tulee myös täyttää *jäljitettävyyss*-tavoite. Käytännössä tämä tarkoittaa sitä, että jokaisen vaatimuksen ja reunaehdon yhteyteen tulee kirjata esittäjätaho ja perustelut. Lisäksi jäljitettävyyss edellyttää, että dokumentaatiosta ilmenee vaatimusten keskinäiset suhteet ja riippuvuudet sekä yhteydet vaatimusdokumentaation ulkoisiin elementteihin kuten testitapauksiin tai arkkitehtuurikuvauksiin.

Katselmoinnissa myös tarkistetaan, onko vaatimukset ja oletukset kirjattu niin yksiselitteisesti, ettei niitä voi tulkita väärin. Lisäksi varmistetaan, että keskeiset termit on määritelty sanastoon ja että termien käyttö on johdonmukaista läpi dokumentaation. Näiden ehtojen täytyessä dokumentaatio on saavuttanut *yksiselitteisyyden* laatutavoitteen. Yk-

siselitteisyyden ohella pyritään täyttämään *ymmärrettävyys*-tavoite. Se edellyttää, että kaikki vaatimukset, reunaehdot ja oletukset on kuvattu niin, että kaikki niiden parissa työskentelevät ymmärtävät ne samalla tavalla ja oikein. Olisi erittäin hyödyllistä, että katselmointiin osallistuu myös teknisten suunnitelmien laatijoita ja arkkitehteja. Heidän tulee pystyä laatimaan vaatimusmäärittelyyn pohjautuvat suunnitelmat toteutustyyliä, joten heidän pitää ymmärtää kirjatut vaatimukset täydellisesti. Katselmoinnissa he voivat tarvittaessa pyytää tarkennuksia vaatimukseen, jos jokin asia vaikuttaa epäselvältä tai vaikeasti ymmärrettävältä.

Vaatimusten tulee myös olla määritelty ja kirjattu niin tarkasti, että niitä voidaan vertailla ja arvioida, toteutuuko jokin vaatimus vai ei. Näin on mahdollista täyttää *mitattavuuden* laatutavoite. Liian ympäröivät vaatimukset muokataan konkreettiseen, mitattavaan muotoon. Vaatimukset käydään läpi myös *asianmukaisuuden* laatutavoitteen näkökulmasta. Sen mukaisesti vaatimusten tulee liittyä tulevan järjestelmän tavoitteisiin ja ongelmakenttään. Mahdolliset toteutustapaan tai toteutuksen yksityiskohtiin kohdistuvat vaatimukset poistetaan. Lisäksi arvioidaan vaatimusten *kelpoisuus*-tavoitetta eli vaatimukset katselmoidaan suhteessa annettuun budjettiin, aikatauluun ja käytettävissä olevaan teknologiaan. Avainasemassa on budjetin ja aikataulun määräävä tilaaja ja teknologiasta vastaava arkkitehti. Heidän näkemystensä perusteella vaatimuksia muokataan tai mahdollisesti muutetaan aikataulua tai budjettia.

Vaatimukset myös verifioidaan eli varmistetaan, ettei vaatimusmäärittelydokumentaatioissa ole keskenään ristiriitaisia vaatimuksia. Lisäksi vaatimusten tulee olla linjassa dokumentoitujen oletusten ja rajoitteiden kanssa, jolloin vahvistettu dokumentaatio täyttää *yhdenmukaisuuden* laatutavoitteen. Verifiointin apuna voidaan hyödyntää muun muassa tarkistuslistoja. Lopuksi myös tarkastetaan, että edellisessä iteraatiovariaatiossa varmistetut *muunneltavuus*- ja *hyvä rakenne* -laatutavoitteet täyttyvät edelleen. Havaitut epäkohdat korjataan, ja iteraatiovariaation lopputuloksena on vaatimusdokumentista ja sen liitteistä versio 2.0.

Käytettävästä systeemityömenetelmästä ja projektista riippuen päivitetyn vaatimusdokumentin pohjalta voidaan jo tässä vaiheessa laatia alustavia teknisiä suunnitelmia ja arkkitehtuurihahmotelmia. Arkkitehtuurikuvausten osalta on syytä tiedostaa, että liiketoimintaprosessin kuvastuminen arkkitehtuurissa on eräs vaatimusmäärittelyn onnistu-

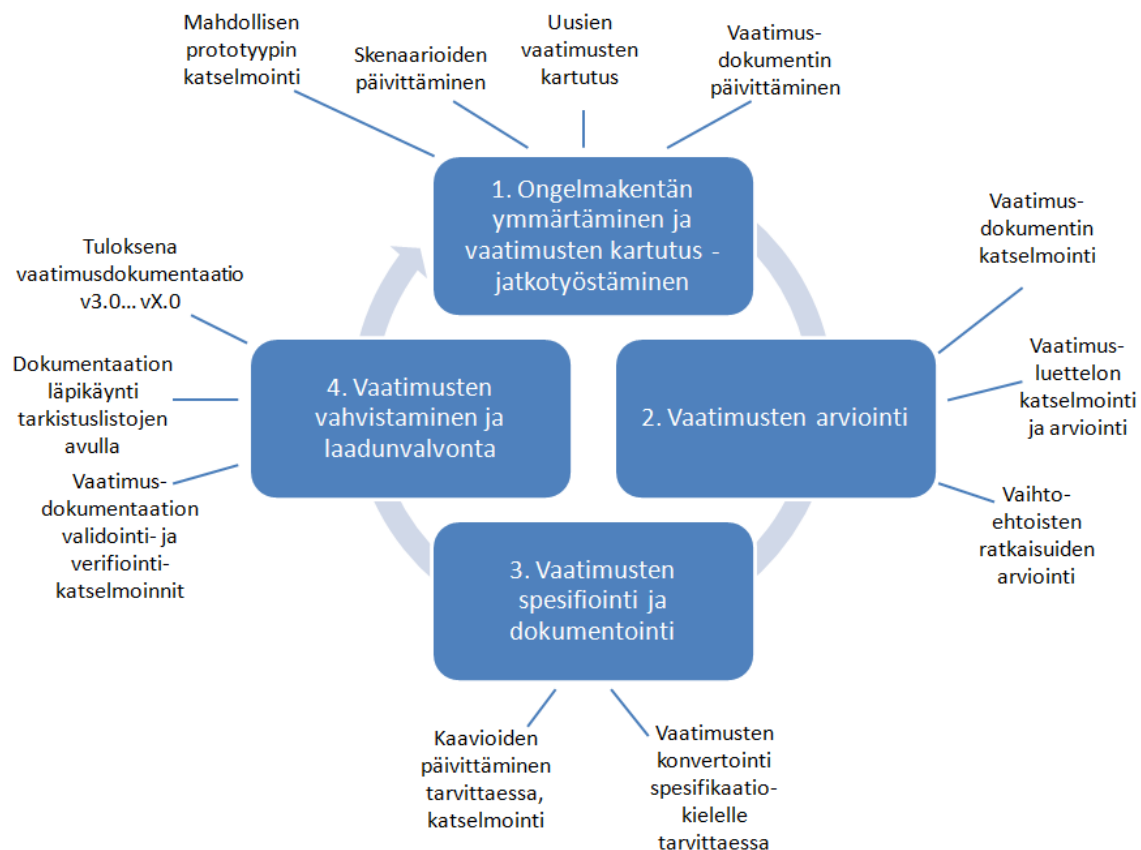


mistekijä [ELM95, 208-209]. Mitä paremmin tulevan järjestelmän arkkitehtuuri saadaan tukemaan toiminnan taustalla olevia liiketoimintaprosesseja, sitä varmemmin tuleva järjestelmä myös täyttää käyttäjien tarpeet. Sähköisen lääkemääräyksen tapauksessa on esimerkiksi täytynyt huomioida sekä lääkemääräysten kirjoittamis- että toimitusprosessit ja suunnitella arkkitehtuuri tukemaan niitä. Arkkitehtuurisuunnittelun lopputuloksena oli olemassa olevia potilas- ja apteekkitietojärjestelmiä hyödyntävä integraatoratkaisu, joka mahdollistaa sekä lääkemääräysten kirjoittamisen että toimittamisen (katso kuva 8).

Arkkitehtuurikuvausten ja teknisten suunnitelmien ohella vaatimusdokumenttia voidaan mahdollisesti myös käyttää jo tässä vaiheessa karkeiden prototyyppien lähtömateriaalina. Täydentävät mallit ja prototyypit ovat eräs Hofmannin ja Lehnerin [HoL01, 65] esille nostamista vaatimusmäärittelytyön parhaista käytännöistä. Niiden avulla on mahdollista esitellä tilaajalle tulevan järjestelmän toimintaa ja ominaisuuksia riittävän konkreettisella tasolla palautteen saamiseksi. Samalla voidaan varmistua siitä, että käyttäjät ymmärtävät, mihin tulevassa ratkaisussa pyritään [ELM95, 208-209].

#### ***4.4 Vaatimusmäärittelyn kolmas iteraatiovariaatio***

Kolmas iteraatiovariaatio keskittyy vaatimusten jatkotyöstämiseen ja tarkentamiseen. Se voidaan toistaa tarpeen mukaan yhden tai useamman kerran. Jos edellisessä iteraatiossa tunnistettiin uusia sidosryhmiä, niiden edustajat kutsutaan mukaan tähän iteraatioon. Heidän perehdytetään kertomalla vaatimusmäärittelyprosessista ja esittelemällä tähän mennessä tuotettu dokumentaatio. Kolmannen tason iteraatiovariaatioiden sijoittuminen kehitysprojektissa voi vaihdella valitusta systeemyömenetelmästä riippuen; asiaa käsitellään tarkemmin luvussa 4.5. Kolmannen iteraatiovariaation mukaiset toimenpiteet on esitelty kuvassa 11.



Kuva 11. Vaatimusmäärittelymallin kolmannen iteraatiovariaation toiminnot.

### *Iteraatiovariaatio 3 - ongelmakentän ymmärtäminen ja vaatimusten kartutus*

Kuvan 11 mukaisesti kolmas iteraatiovariaatio aloitetaan katselmoimalla prototyyppiä, mikäli sellainen edellisen iteraation päätteeksi on tuotettu. Yksinkertaisimmillaan kyseessä voi olla edellisen iteraation päätteeksi paperille mallinnettu käyttöliittymäprototyyppi, jonka avulla käydään läpi tulevan järjestelmän ulkoasua ja näytöjä. Kehitystyön edetessä tulevissa vaatimusmäärittelyiteraatioissa prototyypit voivat olla tätä monimutkaisempia, esimerkiksi järjestelmän osasovelluksia. Järjestelmän käyttäjien on usein helpompi kommentoida ja arvioida jotakin konkreettista tuotosta kuin yrittää hahmottaa asioita mielikuvituksessaan. Prototyyppien avulla sidosryhmiltä pyritäänkin saamaan varhaista palautetta tulevaan järjestelmään sisällytettävistä ominaisuuksista ja toiminnoista; toimiiko kaikki toivotulla tavalla ja onko ulkoasu toivotunlainen. Prototyypin pohjalta voi nousta esille uusia vaatimuksia tai rajoitteita, ja ne kirjataan ylös vaatimusluetteloon. Myöhemmissä kolmannen tason iteraatioissa prototyypin avulla voidaan

varmistaa, että osallistujat ymmärtävät, mitä tuleva järjestelmä tekee ja ei tee, mikä oli yksi keskeisistä vaatimusmäärittelyn onnistumistekijöistä [ELM95, 208-209].

Kolmannessa iteraatiovariaatiossa myös tarkastetaan aiemmin mallinnetut skenaariot, onko niihin tullut jotakin päivitystarvetta. Työn edetessä on saatettu esimerkiksi tunnistaa uusia poikkeuksia johonkin normaaliskenaarioon. Sähköisen lääkemääräyksen tapauksessa olisi voitu oivaltaa, että järjestelmässä täytyy voida varautua siihen, etteivät yhteydet eri osajärjestelmien kesken toimikaan. Asiakkaan on saatava hänen terveydelleen kriittinen lääke yhteysongelmista huolimatta, joten on mietittävä, miten tähän poikkeukseen varaudutaan. Skenaarioita päivitetään tarpeen mukaan, ja mahdolliset uudet vaatimukset lisätään vaatimusluetteloon. Vaatimusdokumenttia päivitetään myös muilta osin tarvittaessa vastaamaan nykyistä ymmärrystä.

### *Iteraatiovariaatio 3 - vaatimusten arviointi*

Vaatimusten arviointi suoritetaan kolmannessa iteraatiovariaatiossa katselmoimalla vaatimusdokumentti ja vaatimusluettelo yhteisesti kaikkien sidosryhmäedustajien kanssa. Lopuksi vaatimukset priorisoidaan tarvittaessa uudelleen. Vaikka uusia vaatimuksia ei olisikaan kyseisessä iteraatiossa tunnistettu, on niiden keskinäinen tärkeysjärjestys voinut muuttua. Lisäksi katselmoidaan ja priorisoidaan ratkaisuvaihtoehdot, jos niihin liittyen on selvinnyt uusia asioita. Neuvotteluita jatketaan tarvittaessa niin kauan, että priorisointi saadaan tehtyä ja osallistujien kesken vallitsee yhteisymmärrys suositellusta ratkaisusta.

### *Iteraatiovariaatio 3 - vaatimusten spesifointi ja dokumentointi*

Vaatimusten spesifointi ja dokumentointi ei tässä iteraatiovariaatiossa poikkea aiemmista iteraatiovariaatioista. Vaatimukset siis konvertoidaan määrätylellä spesifikaatiokielelle, jos siihen on päädytty, ja kaaviot päivitetään sekä katselmoidaan tarvittaessa.

### *Iteraatiovariaatio 3 - vaatimusten vahvistaminen ja laadunvalvonta*

Kolmannessa iteraatiovariaatiossa vaatimusten vahvistaminen ja laadunvalvonta suoritetaan samoin kuin toisessa iteraatiovariaatiossa. Yhteiskatselmoinnissa vaatimusdokumentaatio käydään läpi van Lamsweerden [Lam09, 35-36] määrittelemien vaatimusdokumenttien laatutavoitteiden näkökulmasta. Seuraavassa vielä lyhyt kertaus laatutavoitteista, jotka esiteltiin tarkemmin toisen iteraatiovariaation vaatimusten vahvistaminen ja

laadunvalvonta -vaiheessa. Vaatimusten tulee siis olla sopivia sidosryhmien tarpeisiin ja asianmukaisia liittyen nimenomaan tulevan järjestelmän tavoitteisiin ja ongelmakenttään. Niiden on oltava yksiselitteisiä ja ymmärrettäviä, ja niiden on muodostettava yhdenmukainen, järkevä kokonaisuus. Vaatimusten tulee olla kelpoisia suhteessa budjettiin, aikatauluun ja käytettävissä olevaan teknologiaan, ja niiden toteutumista on voitava mitata. Vaatimusten on oltava jäljitettävissä niiden esittäjätahoon, ja niiden keskinäisten suhteiden ja riippuvuuksien on tultava ilmi dokumentaatiosta. Lisäksi vaatimusdokumentaation tulee noudatella hyvää rakennetta ja olla helposti muunneltavissa.

Uutena dokumentaation laatutavoitteena kolmannessa iteraatiovariaatiossa tarkastellaan *täydellisyys*-tavoitetta. Se edellyttää, että vaatimusmäärittelyssä on pystytty tunnistamaan kaikki tulevaan järjestelmään kohdistuvat tarpeet ja luomaan niiden pohjalta tavoitteet tulevalle järjestelmälle. Lisäksi kaikki mahdolliset virhe- ja poikkeustilanteet on pitänyt pystyä tunnistamaan ja asettamaan niiden ratkaisemiseksi tarvittavat vaatimukset. Jos katselmoinnissa herää epäily siitä, ettei kaikkia vaatimuksia ole tunnistettu tai että jokin poikkeama tai mahdollinen virhetilanne on käsittelemättä, on turvallisinta kirjata epäily yksityiskohtineen tiedoksi ja palata siihen uudessa kolmannen tason iteraatiovariaatiossa. Erityisesti turvallisuuskriittisten, kuten sähköisen terveydenhuollon tietojärjestelmien kehitysprojekteissa täydellisyystavoitteeseen kannattaa suhtautua vakavasti. Kun potilaiden terveys ja jopa elämä riippuu tietojärjestelmän toiminnasta, ei puolivalmiina julkaistaviin tuotteisiin ole yksinkertaisesti varaa. Sen vuoksi viimeistään viimeisessä vaatimusmäärittelyiteraatioissa ennen valmiin tuotteen julkaisua tulisi täydellisyystavoitteen olla täytetty.

Kun kolmas iteraatiovariaatio käydään läpi ensimmäisen kerran, on tuloksena vaatimusdokumentaatio v3.0. Kun kolmatta iteraatiovariaatiota sitten toistetaan myöhemmin, kasvatetaan hyväksytyn vaatimusdokumentaation versiointia aina yhdellä: 4.0... n.0. Näin tiedetään, monennessako vaatimusmäärittelyiteraatioissa kyseinen dokumentaatio on tuotettu. Kaikkien kolmannen tason vaatimusmäärittelyiteraatioiden päätteeksi valitusta systeemityömenetelmästä riippumatta olisi hyvä tuottaa prototyyppisiä ja arkkitehtuurisuunnitelmia. Niiden avulla vaatimukset ja vaatimusten toteutettavuus konkretisoidaan niin sidosryhmille kuin kehitystiimillekin. Seuraavissa iteraatioissa tuotoksia voidaan katselmoida ja tarkentaa niin vaatimuksia kuin suunnitelmiakin saadun palautteen mukaisesti.

#### ***4.5 Iteraatiovariaatioiden sijoittuminen erilaisissa kehitysprojekteissa***

Vaatimusmäärittelymallia voidaan soveltaa erilaisissa tietojärjestelmäprojekteissa riippumatta siitä, mitä systeemityömenetelmää siinä noudatetaan. Seuraavaksi esitellään iteraatiovariaatioiden sijoittuminen vesiputousmallilla ja Scrumilla toteutetuissa kehitysprojekteissa. Kuvattujen periaatteiden pohjalta vaatimusmäärittelymalli on sovellettavissa myös muihin systeemityömenetelmiin.

Liitteessä 4 on esitelty vesiputousmalli ja vaatimusmäärittelyn iteraatiovariaatioiden sijoittuminen siinä. Kuten teoriaosuudessa luvussa 2.2 todettiin, vesiputousmallissa vaatimusmäärittelytyö suoritetaan määrittelyvaiheen aikana. Kuten liitteestä 4 ilmenee, vaatimusmäärittelyn ensimmäinen ja toinen iteraatiovariaatio (VM1 ja VM2) suoritetaan kaikkine työvaiheineen kertaalleen määrittelyvaiheen alussa. Niiden lisäksi määrittelyvaiheessa suoritetaan kolmas iteraatiovariaatio (VM3) yhden tai useamman kerran. Toistamistarve riippuu siitä, miten kattavalle tasolle vaatimukset on saatu dokumentoitua, sillä niiden pohjalta on suunnitteluvaiheessa pystyttävä laatimaan tarkat tekniset suunnitelmat järjestelmän toteuttamiseksi. Erityisesti sähköisen terveydenhuollon tietojärjestelmien kohdalla on hyvä pitää mielessä täydellisyyden laatutavoite. Sen mukaisesti kaikki mahdolliset vaatimukset tulee olla tunnistettu ja kuvattu viimeisen kolmannen tason iteraatiovariaation päätteeksi.

Lisäksi kolmatta iteraatiovariaatiota voidaan joutua suorittamaan vesiputousmallissa myöhemminkin, mikäli havaitaan tarvetta muokata olemassa olevia vaatimuksia. Tällainen tilanne voi syntyä esimerkiksi toteutusvaiheessa, jos todetaan, ettei järjestelmää pystytä syystä tai toisesta toteuttamaan määritysten mukaisesti. Vastaavanlainen tilanne voi tulla eteen testausvaiheessa, jos todetaan, ettei tuleva järjestelmä täytäkään tilaajan tarpeita. Tällöin joudutaan palaamaan takaisin määrittelyvaiheeseen ja suorittamaan VM3 niin monta kertaa, kuin määritysten tarkentaminen edellyttää. Tämän jälkeen jatketaan normaalisti vesiputousmallin vaiheita eteenpäin yksi kerrallaan. Palaaminen takaisin määrittelyvaiheeseen on kuitenkin aikaa vievää, tuottaa lisää työtä ja kuluttaa resursseja. Sen vuoksi kannattaa ennemmin toistaa kolmatta iteraatiovariaatiota jo alkuvaiheessa riittävän monta kertaa, jotta vaatimukset saadaan selville riittävän kattavasti.

Ketterissä projekteissa vaatimusmäärittelyä voidaan tehdä vesiputousmallia joustavammin kehitysprojektin eri vaiheissa eli kehitysiteraatioissa. Vaatimusmäärittelyn iteraa-

tiovariaatioiden sijoittumista Scrum-mallissa on esitelty kuvassa 12. Sen mukaisesti ensimmäinen ja toinen iteraatiovariaatio (VM1 ja VM2) suoritetaan kertaalleen ensimmäisen kehitysiteraation alkupelin aikana. Tämän jälkeen voidaan kyseisen kehitysiteraation kehitysvaiheessa suunnitella ja toteuttaa alustaviin vaatimuksiin pohjautuva julkaisuhahmotelma eli prototyyppi. Prototyypin avulla tilaajalle voidaan esitellä tulevaa järjestelmää ja sen ominaisuuksia kehitysiteraation päätteeksi. Tilaajalta saatua palautetta käytetään lähtökohtana toisessa kehitysiteraatiossa käynnistettävälle vaatimusmäärittelyn kolmannelle iteraatiovariaatiolle (VM3).

Kehitys- iteraatiot	Scrumin kehitysiteraatioiden vaiheet		
Kehitys- iteraatio 1	Alkupeli VM1 VM2	Kehitysvaihe	Loppupeli
Kehitys- iteraatio 2	Alkupeli VM3	Kehitysvaihe	Loppupeli
Kehitys- iteraatio 3	Alkupeli VM3	Kehitysvaihe	Loppupeli
Kehitys- iteraatio 4...n	Alkupeli VM3	Kehitysvaihe	Loppupeli

Kuva 12. Vaatimusmäärittelyn iteraatiovariaatioiden VM1, VM2 ja VM3 sijoittumisehdotus Scrum-mallissa.

Lisäksi kolmas iteraatiovariaatio (VM3) voidaan Scrum-mallissa suorittaa joustavasti kehitysprojektin myöhemmissä kehitysiteraatioissa aina tarvittaessa. Tällöin vaatimusten tarkistus ja tarkennus voisi tapahtua esimerkiksi aina uuden kehitysiteraation alkupelin aikana käymällä läpi kolmas iteraatiovariaatio. Näin saadaan varmistettua ajan tasalla olevat vaatimukset, joita käytetään pohjana kyseisessä kehitysiteraatiossa laadittaville teknisille suunnitelmille sekä suoritettavalle toteutus- ja testaustyölle.

Edellä esiteltyjen soveltamisperiaatteiden mukaan mallia voidaan hyödyntää myös muissa systeemityömenetelmissä. Scrumin yhteydessä esitellyt periaatteet ovat sovellettavissa kaikkiin ketteriin, iteratiivisiin menetelmiin, ja vesiputousmalli soveltuu vaiheistettuun työskentelyyn. Käytännössä ensimmäinen ja toinen iteraatiovariaatio suoritetaan systeemityömenetelmästä riippumatta aina peräkkäin vaatimusmäärittelytyön alkuvai-

heessa. Keskeisin soveltamiseen liittyvä seikka on siis se, miten kolmas iteraatiovariaatio ajoittuu kehitystyössä.

## **5 Sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittely**

Maailman väestön keski-ikä nousee ja terveydenhuoltopalveluiden kysyntä kasvaa kiihtyvällä tahdilla. Jotta kysyntään pystyttäisiin vastaamaan, on perinteistä terveydenhuoltoa ollut pakko kehittää joustavampaan, yksilöllisempään ja kustannustehokkaampaan suuntaan. Avuksi on valjastettu ICT-teknologia ja sen saavutukset. Sähköinen terveydenhuolto on kehittynyt viimeisen vuosikymmenen aikana suuresti tarjoten uusia sovelluksia niin terveydenhuollon organisaatioille kuin kuluttajillekin [Thü15, 13]. Kehityksen arvellaan jatkuvan edelleen, ja eräiden arvioiden mukaan vuonna 2017 tullaan maailmassa käyttämään noin 18 miljardia euroa terveydenhuoltoalan ICT-kustannuksiin.

Myös Maailman terveysjärjestö WHO on omalta osaltaan pyrkinyt edesauttamaan sähköisen terveydenhuollon sovelluksien kehittämistä ja käyttöönottoa [Bui06]. Se on kehottanut jäsenvaltioita laatimaan pitkän aikavälin strategioita sähköisten terveydenhuoltopalveluiden kehittämiseksi ja käyttöönottamiseksi. Lisäksi se on mukana kehittämässä standardeja, linjauksia ja ohjeistuksia, joiden avulla olisi mahdollista kehittää entistä toimivampia ja turvallisempia sovelluksia sekä helpottaa niiden käyttöönottoa.

Tämän pro gradu -tutkielman yhtenä tutkimuskysymyksenä oli, mitä erityispiirteitä sähköisen terveydenhuollon tietojärjestelmissä on. Selvisi, että sähköisen terveydenhuollon tietojärjestelmissä useita erityispiirteitä, jotka tulee huomioida niin kehitystyön aikana kuin valmiissa sovelluksissa. Eräs keskeinen erityispiirre on turvallisuuskriittisyys [Kni02, 547]. Se tarkoittaa sitä, että tietojärjestelmän toimintahäiriöt voivat vahingoittaa ympäristöään ja käyttäjiänsä ja johtaa pahimmassa tapauksessa kuolemaan. Konkreettinen esimerkki tästä on sydämentahdistin. Toinen erityispiirre on korostunut tarve tietosuojalle ja -turvalle tietojärjestelmiin tallennettujen ja niissä käsiteltävien potilaiden sensitiivisten tietojen vuoksi [BrH15, 134]. Tietojärjestelmässä käsiteltävien potilastietojen tulee olla suojattuja potilaan yksityisyydensuojan turvaamiseksi. Kolmas erityispiirre on tarve yhteentoimivuudelle muiden keskeisten tietojärjestelmien kanssa [BLC15, 95-97, 110]. Jotta tietoja voitaisiin välittää ongelmitta eri organisaatioiden kesken, tulee

järjestelmien olla yhteentoimivia. Keskeinen tekijä tässä ovat standardit, jotka mahdollistavat esimerkiksi toimivan sanomavälityksen.

Neljäs erityispiirre sähköisen terveydenhuollon tietojärjestelmissä on voimakas sidonnaisuus lakeihin ja asetuksiin [PKK15, 48-49]. Terveydenhuolto on tarkkaan säädelty toimiala, jossa on otettava huomioon paikalliset terveydenhuollolle spesifisti säädetyt lait ja asetukset. Lisäksi on tapauskohtaisesti selvitettävä, mitä muita lakeja, kuten tietosuojaan tai tuoteturvallisuuteen liittyviä lakeja tulee kehitysprojektissa huomioida. Viidentenä erityispiirteenä voidaan pitää sidosryhmien moninaisuutta ja heterogeenisyyttä [VFS15, 157-159]. Sidosryhmät voivat olla järjestelmän käyttäjätahoja, rahoittajia tai poliittisia päättäjiä. Lisäksi sähköisen terveydenhuollon tietojärjestelmän tilaaja ei välttämättä ole järjestelmän käyttäjä, mikä voi tuoda omat haasteensa kehitystyöhön ja sen tavoitteisiin. Tarve jakaa ja siirtää tietoja eri organisaatioiden ja toimijoiden kesken edellyttää, että myös ne tunnistetaan sidosryhmiksi ja että niiden kanssa tehdään tiivistä yhteistyötä yhteentoimivuuden takaamiseksi.

Tämän tutkielman puitteissa tunnistettiin siis viisi keskeistä erityispiirrettä. Voi hyvin olla, että tarkemmat analyysit paljastavat vielä uusia erityispiirteitä. Eräs jatkotutkimuksen aihe voisikin olla analyysin jatkaminen ja sen myötä tunnistettavien erityispiirteiden integrointi vaatimusmäärittelymalliin.

Edellä esiteltyt erityispiirteet on huomioitava jo tietojärjestelmän kehitysvaiheessa, jotta valmis tietojärjestelmä olisi toimiva ja turvallinen. Tämän vuoksi keskeiseen roolin tietojärjestelmän kehitysprosessissa nousee vaatimusmäärittelytyö. Yksi tutkielman tutkimuskysymyksistä koski nimenomaan vaatimusmäärittelyä: mikä asema vaatimusmäärittelyllä on tietojärjestelmäkehityksessä. Teoriaosuudessa todettiin, että vaatimusmäärittelyn aikana selvitetään, mitä toiminnallisuuksia tulevassa järjestelmässä pitää olla ja miten sen tulee toimia [Lam09, 42-43]. Tunnistetut vaatimukset toimivat pohjana useille kehitysprojektin aktiviteeteille, kuten projektiarvioille, tarjouspyynnöille, työsuunnitelville, ohjeistuksille, arkkitehtuureille, prototyypeille, testitapauksille ja niin edelleen. Ne toimivat siis pohjana suunnittelu- ja toteutustyölle, ja niitä vasten todennetaan, toimiiko järjestelmä niin kuin sen olisi tarkoitus toimia. Tutkimuksissa on todettu, että puuttuvat tai virheellisesti määritellyt vaatimukset eli epäonnistunut vaatimusmäärittely on yleisin syy tietojärjestelmien kehitysprojektien epäonnistumiseen [The95, 4-5 ja



FGZ15, 26]. Kiistämättä on siis selvää, että vaatimusmäärittelytyö on tietojärjestelmien kehittämisen selkäranka. Ilman sitä on lähes mahdotonta tuottaa toimivia, käyttäjien tarpeet täyttäviä tietojärjestelmiä. Huolellinen ja kattava vaatimusmäärittely on siis erittäin tärkeää sähköisen terveydenhuollon tietojärjestelmien kehityksessä.

Tässä tutkielmassa tutustuttiin sähköiseen lääkemääräykseen sähköisen terveydenhuollon esimerkkitapauksena. Selvisi, ettei Suomen sähköisen lääkemääräyksen vaatimusmäärittelyn suorittamisesta ole saatavilla materiaalia. Myöskään maailmalta ei löytynyt esimerkkejä tai ohjeistuksia siitä, kuinka tällaisten turvallisuuskriittisten, lakiperusteisten, monimutkaisten sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittely kannattaisi suorittaa. Yhtenäinen, standardoitu malli helpottaisi vaatimusmäärittelyn suorittamista, kun jokaisessa kehitysprojektissa ei tarvitse etsiä ja miettiä sopivia käytäntöjä yrityksen ja erehdyksen kautta. Lisäksi malli parantaa vaatimusmäärittelyn tehokkuutta ja laatua; se ohjeistaa, mitä tulee tehdä ja huomioida missäkin vaiheessa, jolloin voidaan välttyä unohduksilta ja virheiltä. Samalla yhtenäiset vaatimusmäärittelytyön toimintatavat helpottavat henkilöiden siirtymistä projektista toiseen, kun työvaiheet ja niiden sisällöt ovat tuttuja ja ennakoitavissa.

Näistä lähtökohdista nousi kolmas tutkielmaa ohjaava tutkimuskysymys: miten sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittely kannattaa suorittaa. Kysymykseen vastaa tutkielman kontribuutio-osuudessa koostettu, kolmesta erilaisesta iteraatiovariaatiosta koostuva vaatimusmäärittelymalli (katso liite 3). Malli pohjautuu van Lamsweerden kuvaamaan iteratiiviseen vaatimusmäärittelyprosessiin [Lam09, 34-35]. Prosessi sisältää neljä työvaihetta: ongelmakentän ymmärtämis- ja vaatimusten kartutusvaiheen, vaatimusten arviointivaiheen, vaatimusten spesifiointi- ja dokumentointivaiheen sekä vaatimusten vahvistamis- ja laadunvalvontavaiheen. Kunkin vaiheen tuloksena on edellistä kehittyneempi versio tulevaan järjestelmään kohdistuvista vaatimuksista. Kaikki työvaiheet toistetaan mallin jokaisessa iteraatiovariaatiossa, mutta niiden painotukset ja sisällöt vaihtelevat. Sähköisen terveydenhuollon järjestelmien erityispiirteiden vuoksi iteratiivinen lähestymistapa vaatimusmäärittelyssä on järkevä toimintatapa, sillä sen myötä vaatimuksia voidaan tarkentaa asteittain ymmärryksen kasvaessa.

Mallin ensimmäisessä iteraatiovariaatiossa keskitytään erityisesti tilaajan näkökulmaan ja toisessa iteraatiovariaatiossa sidosryhmien näkökulmaan. Nämä iteraatiovariaatiot on tarkoitus käydä läpi kertaalleen vaatimusmäärittelytyön alkuvaiheessa. Kolmannen iteraatiovariaation tavoitteena on työstää tunnistettuja vaatimuksia vielä tarkemmalle tasolle ja tunnistaa mahdollisia uusia vaatimuksia. Se voidaan toistaa kehitysprojektin aikana niin monta kertaa, että vaatimukset on saatu määriteltyä kattavasti ja riittävän tarkalle tasolle. Valitusta systeemyömenetelmästä riippuu, missä vaiheessa kehitysprosessia kolmatta iteraatiovariaatiota käytetään. Vesiputousmallissa se sisältyy aina määrittelyvaiheeseen, kun taas esimerkiksi Scrumissa se voidaan suorittaa tarpeen mukaan vaikka jokaisessa kehitysiteraatioissa.

Vaatimusmäärittelymallissa on huomioitu edellä mainitut sähköisen terveydenhuollon erityispiirteet. Lisäksi siinä sovelletaan van Lamsweerden [Lam09, 35-36] määrittelemiä vaatimusdokumentaation laatutavoitteita, El Emamin ja Madhavjin [ELM95, 208-209] tunnistamia vaatimusmäärittelyn onnistumistekijöitä sekä Hofmannin ja Lehnerin [HoL01, 65-66] esittelemiä vaatimusmäärittelyn parhaita käytäntöjä.

Mallin mukaisesti suoritettun vaatimusmäärittelytyön avulla sähköisen terveydenhuollon tietojärjestelmien kehitysprojekteilla on entistä paremmat edellytykset onnistua ja tuottaa käyttäjien tarpeet täyttävä tietojärjestelmä. Keskeistä on se, että kehitysprojekteissa kannattaa jo esiselvitysvaiheessa miettiä mallin soveltaminen; mitkä laatutavoitteista ja onnistumistekijöistä ovat olennaisia juuri kyseisessä projektissa. Valitut tavoitteet ja tekijät toimivat ohjenuorana projektissa tehtäville päätöksille ja panostukselle. Kehitysprojektin luonteesta tai tavoitteesta riippumatta tulee silti muistaa, että vaatimusmäärittelyyn on aina varattava riittävästi aikaa ja resursseja. "Riittävästi" on toki suhteellinen käsite, ja sitä määrittävät omalta osaltaan laatutavoitteisiin ja onnistumistekijöihin liittyvät valinnat.

Vaikka malli tämän tutkielman yhteydessä onkin esitelty erityisesti sähköisen terveydenhuollon tietojärjestelmien näkökulmasta, se on sovellettavissa myös muiden alojen tietojärjestelmien vaatimusmäärittelyyn. Sen käytännöt ovat monistettavissa ja skaalattavissa erikokoisiin projekteihin käytetystä systeemyömenetelmästä riippumatta. Iteratiivisen luonteensa vuoksi malli saattaa kuitenkin toimia paremmin iteratiivisuuteen pohjautuvissa ketterissä menetelmissä.

Mallin soveltamisesta erilaisissa kehitysprojekteissa tarvittaisiin käyttökokemuksia, joiden perusteella sitä voitaisiin kehittää vielä toimivammaksi. Esimerkiksi El Emamin ja Madhavjin onnistumistekijöitä [ELM95, 208-209] voisi olla tarpeen viedä vielä konkreettisemmalle ja mitattavammalle tasolle ja integroida tulokset tähän malliin. Näin kehitysprojektit saisivat tarkempia viitteitä siitä, miten tehdyt valinnat vaikuttavat projektien onnistumiseen. Ajatellaan, että kehitysprojektissa olisi valittu keskeiseksi onnistumistekijäksi M18 (missä laajuudessa avainkysymykset on pystytty ratkaisemaan). Tekijä ei sinällään vielä kerro, miten suuri määrä avainkysymyksistä pitää olla ratkaistu missäkin vaiheessa projektia. Olisi hyvä, että tästä olisi käytettävissä jonkin mittaluku, kuten: onko ensimmäisen 3-tason iteraatiovariaation jälkeen 75 % avainkysymyksistä pystytty ratkaisemaan? Näin tiedetään, mitä tasoa olisi tarpeen tavoitella. Lisäksi malliin voisi olla hyödyllistä lisätä enemmän tietoa standardeista ja IHE:n laatimista profiileista [Kni02, 549-550].

Tässä tutkielmassa keskityttiin erityisesti sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittelyyn ja laadittiin sitä tukeva malli. Vaikka vaatimusmäärittely onkin tietojärjestelmien kehitystyön selkäranka, on myös muilla kehitystyön vaiheilla merkitystä. Voisikin olla hyödyllistä laajentaa mallia siten, että siinä käsiteltäisiin kaikki kehitystyön vaiheet. Näin ollen siinä olisi oma osuutensa esiselvitykselle, vaatimusmäärittelylle, suunnittelulle, toteutukselle, testaukselle, käyttöönotolle ja ylläpidolle. Jokaisessa osuudessa kuvattaisiin, miten vaihe kannattaa suorittaa ja mihin asioihin tulee kiinnittää huomiota. Läpileikkaavia elementtejä voisivat olla esimerkiksi roolit ja standardit, jotka sisältyvät kaikkiin vaiheisiin.

Toimivat, käyttökelpoiset ja toimivat mallit helpottaisivat sähköisen terveydenhuollon tietojärjestelmien kehitystyötä. Koska kyseessä on maailmanlaajuinen toimintakenttä, olisi mallien kehitys- ja koordinoitavuuden hyvä tapahtua keskitetysti. Keskeisiä toimijoita tässä olisi esimerkiksi Maailman terveysjärjestö WHO sekä alueelliset toimijat kuten Euroopan komissio. Niiden kautta voitaisiin panostaa mallien kehitystyöhön ja koestaa malleja sopivissa hankkeissa. Saatujen kokemusten pohjalta malleja voitaisiin vielä hioa, ennen kuin ne jalkautettaisiin laajempaan käyttöön. Panostus mallien suunnittelu- ja kehitystyöhön tulisi varmasti maksamaan itsensä takaisin aiempaa toimivampien ja turvallisempien tietojärjestelmien muodossa.

## 6 Yhteenveto

Tutkielma käsittelee sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittelyä. Selvisi, että näillä tietojärjestelmillä on tiettyjä erityispiirteitä, jotka tulee huomioida jo kehitysvaiheessa. Sähköisen terveydenhuollon tietojärjestelmät ovat yleensä turvallisuuskriittisiä, eli tietojärjestelmän toimivuus ja toimintahäiriöt vaikuttavat merkittävästi käyttäjän tai asiakkaan terveyteen. Niissä myös käsitellään potilaisiin liittyviä sensitiivisiä tietoja, joten tietosuojan ja -turvallisuuden tulee kiinnittää erityistä huomiota. Käyttöönotettavien järjestelmien tulee olla yhteentoimivia toimialan muiden organisaatioiden ja järjestelmien kanssa, joten niissä on noudatettava sovittuja standardeja. Sähköistä terveydenhuoltoa määrittelevät useat lait ja asetukset, jotka tulee huomioida tietojärjestelmien kehitystyössä. Lisäksi toimialan tietojärjestelmiin liittyy useita sidosryhmiä, joiden tarpeet ja vaatimukset tulee selvittää vaatimusmäärittelyn yhteydessä.

Vaatimusmäärittelyllä onkin erittäin keskeinen merkitys tietojärjestelmien kehitysprojektien onnistumiselle. Vaatimusmäärittelytyössä tuotetut dokumentit toimivat lähdemateriaalina ja tukena useille muille kehitysprojektin aktiviteeteille, kuten suunnittelulle ja testaukselle. Jos vaatimuksia ei ole kartoitettu riittävästi tai ne ovat epäselviä, ei tuleva järjestelmä todennäköisesti täytä käyttäjiensä tarpeita. Sen sijaan huolella tehty vaatimusmäärittely tukee muita kehitysprosessin vaiheita nopeuttaen kehitystyön etenemistä, alentaen kustannuksia ja kasvattaen asiakastyytyväisyyttä.

Sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittelyn suorittamiseen ei ole olemassa mallia tai sapluunaa. Niinpä kehitysprojektit joutuvat keksimään käytänteet itse yrityksen ja erehdyksen kautta. Tässä tutkielmassa laadittiin malli sähköisen terveydenhuollon tietojärjestelmien vaatimusmäärittelyyn. Malli koostuu kolmesta erilaisesta iteraatiovariaatiosta, joiden myötä vaatimukset tarkentuvat ja hioutuvat tarkemmalle tasolle projektin edetessä. Siinä huomioidaan tutkielmassa tunnistetut sähköisen terveydenhuollon erityispiirteet. Taulukosta 2 ilmenee, millaisilla keinoilla tutkielmassa tunnistettuja sähköisen terveydenhuollon erityispiirteitä mallissa käsitellään.

<i>Erityispiirre</i>	<i>Käsittely mallissa</i>
<i>Turvallisuuskriittisyys</i>	<ul style="list-style-type: none"> <li>- kiinnitetään erityistä huomiota ei-toiminnallisten vaatimusten laatuvaatimukseen (esimerkiksi suorituskyky ja luotettavuus)</li> <li>- edellytetään vaatimusdokumentaatiolta täydellisyyttä, eli kaikki mahdolliset vaatimukset ja poikkeustilanteet tulee olla tunnistettu</li> <li>- kiinnitetään tarvittaessa vastaavanlaisten turvallisuuskriittisten järjestelmien asiantuntija projektiin</li> </ul>
<i>Korostunut tarve tietosuojalle ja -turvalle</i>	<ul style="list-style-type: none"> <li>- selvitetään varhaisessa vaiheessa järjestelmässä käsiteltävät tiedot ja niihin liittyvät suojaustarpeet</li> <li>- kiinnitetään erityistä huomiota ei-toiminnallisten vaatimusten laatuvaatimukseen (turvallisuus ja suojaus)</li> </ul>
<i>Yhteentoimivuusvaateet</i>	<ul style="list-style-type: none"> <li>- selvitetään liittymät muihin järjestelmiin, joiden kanssa tulevan järjestelmän täytyy olla yhteentoimiva</li> <li>- selvitetään keskeiset standardit ja noudatetaan niitä läpi kehitysprosessin</li> </ul>
<i>Sidonnaisuus lakeihin ja asetuksiin</i>	<ul style="list-style-type: none"> <li>- selvitetään, mitä lakeja, asetuksia ja määräyksiä noudatettava ja nostetaan niistä vaatimuksia</li> <li>- kiinnitetään toimialan lainsäädännön asiantuntija projektiin</li> </ul>
<i>Moninaiset heterogeeniset sidosryhmät</i>	<ul style="list-style-type: none"> <li>- selvitetään tulevaan järjestelmään liittyvät sidosryhmät ja sitoutetaan niiden edustajat osallistumaan vaatimusmäärittelyyn prosessin alusta loppuun</li> <li>- tehdään osa vaatimusten kartuttamisesta ja arvioinnista soveltuvissa pienryhmissä</li> <li>- tunnistetaan toimialalla meneillään olevat kehityshankkeet ja otetaan ne mukaan sidosryhmiksi, jos niillä on yhteisiä intressejä (esimerkiksi edellytetään yhteentoimivuutta tuotettavien järjestelmien kesken)</li> <li>- priorisoidaan sidosryhmät niiden merkityksellisyyden mukaan</li> </ul>

Taulukko 2. Sähköisen terveydenhuollon erityispiirteiden käsittely vaatimusmäärittelymallissa.

Taulukon 2 mukaisesti esimerkiksi kehitettävän järjestelmän turvallisuuskriittisyys huomioidaan mallissa useilla eri tavoilla. Vaatimusmäärittelyssä kiinnitetään erityistä huomiota ei-toiminnallisten vaatimusten laatuvaatimukseen, jotka määrittelevät tulevan järjestelmän toiminnan puitteita. Turvallisuuskriittisen järjestelmän vaatimusdokumentaation on oltava täydellistä, eli siellä pitää olla tunnistettuna kaikki normaalit ja poikkeavat työkulut ja niihin liittyvät vaatimukset. Lisäksi projektiin kiinnitetään tarvittaessa asiantuntija, jolla on erityiskokemusta vastaavanlaisen turvallisuuskriittisen järjes-

telmän kehittämisestä. Vastaavasti tulevan järjestelmän yhteentoimivuutta muiden järjestelmien kanssa pyritään edistämään siten, että varhaisessa vaiheessa tunnistetaan liittymät järjestelmiin, joiden kanssa tulevan järjestelmän tulee olla yhteentoimiva. Lisäksi selvitetään, millaisia standardeja toimialaan liittyy, ja valitaan niistä projektissa noudatettavat standardit.

Vaatusmäärittelymalliin on myös integroitu vaatimusdokumentaation laatutavoitteita [Lam09, 30-36] sekä vaatimusmäärittelyn onnistumistekijöitä [ELM95, 65-66] ja parhaita käytäntöjä [HoL01, 208-209]. Integroidut elementit on kuvattu taulukossa 3.

<i>Vaatusdokumentaation laatutavoitteet</i>	<i>Vaatusmäärittelyn onnistumistekijät</i>	<i>Vaatusmäärittelyn parhaat käytännöt</i>
Täydellisyys	Varmistetaan, että käyttäjät ymmärtävät, mitä tuleva järjestelmä tekee ja mitä se ei tee	Vaatusmäärittelyn osuus kehitysprojektin kokonaispanostuksesta tulisi olla noin 15 - 30 %.
Sopivuus	Käyttäjät ovat yksimielisiä suositellusta ratkaisusta	Kiinnitetään projektiin riittävästi osaavia henkilöresursseja
Jäljitettävyyys	Varataan riittävän suuri osuus koko projektin kehittämiskustannuksista vaatimusmäärittelyyn	Käytetään valmiita mallipohjia ja esimerkkejä
Yksiselitteisyys	Liiketoimintaprosessi kuvastuu selkeästi arkkitehtuurissa	Osallistetaan asiakkaat ja käyttäjät vaatimusmäärittelyprosessiin alusta loppuun asti
Ymmärrettävyyys	Nykyinen tietojärjestelmä on analysoitu kattavasti	Laaditaan jäljitettävyyssmatriisi, jota ylläpidetään koko projektin ajan
Mitattavuus	Suositeltu ratkaisu sopii yhteen organisaation strategian kanssa	Tunnistaa ja hyödynnetään kaikki mahdolliset vaatimuslähteet.
Asianmukaisuus	Organisaatiolla on kyky tehdä tarvittavia muutoksia suositellun ratkaisun käyttöönottamiseksi	Käytetään katselmointeja, läpikäyntejä ja skenaarioita vaatimusten validoinnissa ja verifioinnissa
Kelpoisuus		Luodaan hyvät suhteet sidosryhmiin
Yhdenmukaisuus		Priorisoidaan vaatimukset
Muunneltavuus		Käytetään malleja ja prototyyppejä palautteen saamiseksi käyttäjiltä
Hyvä rakenne		

Taulukko 3. Vaatusmäärittelymalliin integroidut elementit.

Kuten taulukosta 3 ilmenee, malliin on sisällytetty 11 van Lamsweerden esittelemää vaatimusdokumentaation laatutavoitetta, 7 El Emamin ja Madhavjin määrittelemää vaatimusmäärittelytyön onnistumistekijää sekä 10 Hofmannin ja Lehnerin kuvaamaa parasta käytäntöä. Jokainen niistä omalta osaltaan tukee vaatimusmäärittelyn onnistumista. Todellisessa elämässä voi olla kunnianhimoista pyrkiä huomioimaan ja täyttämään kaikki malliin integroidut tekijät, eikä se aina ole tarpeellistakaan. Keskeisintä on tiedostaa, että nämä tekijät voivat vaikuttaa vaatimusmäärittelyn onnistumiseen tai epäonnistumiseen. Mallia voi soveltaa erilaisiin kehitysprojekteihin ja organisaatioihin valikoidulla laatutavoitteista, onnistumistekijöistä ja parhaista käytännöistä ne, jotka ovat keskeisiä juuri kyseisessä projektissa. Valitut tavoitteet, tekijät ja käytännöt toimivat ohjenuorana projektissa tehtäville päätöksille ja panostuksille. Mallin avulla vaatimusmäärittelytyön oletetaan helpottuvan ja tehostuvan ja sen laadun paranevan.

## Lähteet

- Acl08 A Clinician's Guide to Electronic Prescribing. The eHealth Initiative Foundation, 2008. [http://www.thecimm.org/PDF/Clinician's\\_Guide\\_to\\_e-Prescribing\\_Final\\_100708.pdf](http://www.thecimm.org/PDF/Clinician's_Guide_to_e-Prescribing_Final_100708.pdf). [19.11.2015]
- ASR02 Abrahamsson, P., Salo, O., Ronkainen, J. ja Warsta, J., Agile Software Development Methods. VTT Publications 478. Espoo, 2002. [Myös <http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>, 29.4.2016].
- AtM11 Attarha, M. ja Modiri, N., Focusing on the Importance and the Role of Requirement Engineering. Proceedings of the 4th International Conference on Interaction Sciences: IT, Human and Digital Content, Busan, Korea, elokuu 2011, sivut 181-184. [Myös [http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6014555&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D6014555](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6014555&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D6014555), 21.5.2015].
- BLC15 Bourguard, K., Le Gall, F. ja Cousin, P., Standards for Interoperability in Digital Health: Selection and Implementation in an eHealth Project. Teoksessa Requirements Engineering for Digital Health, Fricker, S. A., Thümmler, C. ja Gavras, A., toimittajat, Springer, Lontoo, 2015, sivut 95-115.
- BrH15 Brost, G. S. ja Hoffmann, M., Identifying Security Requirements and Privacy Concerns in Digital Health Applications. Teoksessa Requirements Engineering for Digital Health, Fricker, S. A., Thümmler, C. ja Gavras, A., toimittajat, Springer, Lontoo, 2015, sivut 133-154.
- Bui06 Building Foundations for eHealth - Progress of Member States; Report of the WHO Global Observatory for eHealth. World Health Organization. WHO Press, Geneve, 2006.
- CaR08 Cao, L. ja Ramesh, B., Agile Requirements Engineering Practices: An Empirical Study. Software, IEEE, 25,1 (2008), sivut 60–67. [Myös [http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4420071&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D4](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4420071&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4)



420071, 16.11.2015].

- Cys02 Cysneiros, L.M., Requirements Engineering in the Health Care Domain. Proceedings of the IEEE Joint International Conference on Requirements Engineering, Essen, Saksa, syyskuu 2002, sivut 350-356. [Myös [http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1048548&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs\\_all.jsp%3Farnumber%3D1048548](http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1048548&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D1048548), 21.9.2015].
- Ele15 Electronic Prescribing. Internetissä julkaistu materiaali, Wikipedia, 2015. [https://en.wikipedia.org/wiki/Electronic\\_prescribing](https://en.wikipedia.org/wiki/Electronic_prescribing). [2.11.2015]
- ELM95 El Emam, K. ja Madhavji, N. H., Measuring the Success of Requirements Engineering Process. Proceedings of the Second IEEE International Symposium on Requirements Engineering, York, Englanti, maaliskuu 1995, sivut 204-211. [Myös <http://ieeexplore.ieee.org.libproxy.helsinki.fi/stamp/stamp.jsp?tp=&arnumber=512562>, 23.12.2015].
- FGZ15 Fricker, S.A., Grau, R., Zwingli, A., Requirements Engineering: Best Practice. Teoksessa Requirements Engineering for Digital Health, Fricker, S. A., Thümmeler, C. ja Gavras, A., toimittajat, Springer, Lontoo, 2015, sivut 25-46.
- HoL01 Hofmann, H.F. ja Lehner, F., Requirements Engineering as a Success Factor in Software Projects. Software, IEEE, 18, 4(2001), sivut 58-66. [Myös <http://ieeexplore.ieee.org.libproxy.helsinki.fi/stamp/stamp.jsp?tp=&arnumber=936219>, 18.12.2015].
- How14 How does e-prescribing work? Internetissä julkaistu materiaali, The Health Resources and Services Administration, 2014. [Myös <http://www.hrsa.gov/healthit/toolbox/HealthITAdoptiontoolbox/ElectronicPrescribing/epreswork.html>, 18.11.2015].
- Kel14 Kelan terminologinen sanasto. Etuuksiin liittyvät käsitteet. 3. laitos. Sanastokeskus TSK, Helsinki, 2014. [http://www.kela.fi/documents/10180/1169690/Kelan\\_terminologinen\\_sanast](http://www.kela.fi/documents/10180/1169690/Kelan_terminologinen_sanast)

- o.pdf/3e9844c1-002f-4571-a4ce-1e3ded93461a. [19.11.2015]
- Kni02 Knight, J. C., Safety Critical Systems: Challenges and Directions. Proceedings of the 24th International Conference on Software Engineering ICSE, New York, USA, toukokuu 2002, sivut 547-550. [Myös <http://ieeexplore.ieee.org.libproxy.helsinki.fi/stamp/stamp.jsp?tp=&arnumber=1007998>, 23.10.2015].
- KuB14 Kumar, G. ja Bhatia, P.K., Comparative Analysis of Software Engineering Models from Traditional Models to Modern Methodologies. Proceedings of the 4th International Conference on Advanced Computing & Communication Technologies (ACCT), Rohtak, Haryana, Intia, helmikuu 2014, sivut 189-196. [Myös <http://ieeexplore.ieee.org.libproxy.helsinki.fi/stamp/stamp.jsp?tp=&arnumber=6783451>, 30.10.2015].
- Kö114 Köli, K., Kanta-asiantuntijan haastattelu liittyen sähköiseen lääkemääräykseen ja sen vaatimusmäärittelyyn. Helsinki, 7.10.2014.
- Lak14a Laki sähköisestä lääkemääräyksestä 2.2.2007/61. [Myös <http://www.finlex.fi/fi/laki/ajantasa/2007/20070061>, 20.11.2015].
- Lak14b Laki sähköisestä lääkemääräyksestä annetun lain muuttamisesta 251/2014. [Myös <http://www.finlex.fi/fi/laki/alkup/2014/20140251>, 11.3.2016].
- Lam09 Van Lamsveerde, A., Requirements Engineering for System Goals to UML Models to Software Specifications. John Wiley & Sons Ltd, Chichester West Sussex, 2009.
- Mik14 Mikä on epSOS? EpSOS-hankkeen esittelysivu Internetissä. European Patients Smart Open Services, 2014. <http://www.epsos.eu/wwwepsouseuomi/mikae-on-epsos.html>. [8.2.2016]
- Näi07 Näin luet reseptiä. Internetissä julkaistu ohjemateriaali, Tohtori.fi, 2007. <http://www.tohtori.fi/?page=0965185&id=7666887>. [9.11.2015]
- Poh05 Pohjonen, R., Tietojärjestelmät. Teoksessa Tietotekniikan peruskirja, 6. laitos 1. painos, Paananen, J. toimittaja, Docendo Oy, Jyväskylä, 2005, sivut 338-359.

- PEC14 Porterfield, A., Engelbert, K. ja Coustasse, A., Electronic Prescribing: Improving the Efficiency and Accuracy of Prescribing in the Ambulatory Care Setting. Online Research Journal, Perspectives in Health Information Management, kevät 2014. [Myös <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3995494/pdf/phim0011-0001g.pdf>, 23.9.2015].
- PKK15 Purtova, N., Kosta, E. ja Koops.B-J., Laws and regulation for Digital Health. Teoksessa Requirements Engineering for Digital Health, Fricker, S. A., Thümmeler, C. ja Gavras, A., toimittajat, Springer, Lontoo, 2015, sivut 47-74.
- Rid07 Ridinger, M. H. T., The Electronic Prescription Conundrum: Why "e-Rx" Isn't so "e-Z". Clinical Pharmacology & Therapeutics, 81, 1(2007), sivut 13-15.
- Roy70 Royce, W. W., Managing the Development of Large Software Systems. Proceedings of IEEE WesCon 26, Los Angeles, USA, elokuu 1970, sivut 1-9.
- ScS14 Schwaber, K ja Sutherland, J., The Definitive Guide to Scrum: The Rules of the Game. 2014. Scrum Alliance. [Myös <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf#zoom=100>, 26.4.2016].
- Sos10 Sosiaali- ja terveysministeriön asetus lääkkeen määräämisestä 1088/2010. Helsinki, sosiaali- ja terveysministeriö, 2.12.2010. [Myös <http://www.finlex.fi/fi/laki/alkup/2010/20101088>, 8.2.2016].
- Säh12 Sähköisen terveydenhuollon toimintasuunnitelma 2012–2020 - innovatiivista terveydenhuoltoa 21. vuosisadalle. Euroopan komission tiedonanto Euroopan parlamentille, neuvostolle, Euroopan talous- ja sosiaalikomitealle ja alueiden komitealle. Bryssel 6.12.2012. [Myös <http://eur-lex.europa.eu/legal-content/FI/TXT/PDF/?uri=CELEX:52012DC0736&from=EN>, 15.1.2016]
- Säh15a Sähköinen lääkemääräys vaatimusmäärittely, yleiskuvaus 30.10.2015 versio 2.8. Internetissä julkaistu vaatimusmäärittelydokumentti, Kansaneläkelaitos,

- Kansallinen Terveysarkisto Kanta, 2015.  
[http://www.kanta.fi/documents/12105/4092129/eResepti\\_Maarittely\\_YleiskuvausV2.8.pdf/f67631ac-30fa-4d4e-a872-44ad90ed1578](http://www.kanta.fi/documents/12105/4092129/eResepti_Maarittely_YleiskuvausV2.8.pdf/f67631ac-30fa-4d4e-a872-44ad90ed1578). [3.2.2016]
- Säh15b Sähköinen resepti. Esittelymateriaali Internetissä, Kansallinen Terveysarkisto Kanta, 2015. <http://www.kanta.fi/eresepti-esittely>. [22.11.2015]
- The95 The CHAOS Report (1994). Raportti, The Standish Group International, 1995.  
[https://www.standishgroup.com/sample\\_research\\_files/chaos\\_report\\_1994.pdf](https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf). [23.10.2015]
- Thü15 Thümmeler, C., Digital Health. Teoksessa Requirements Engineering for Digital Health, Fricker, S. A., Thümmeler, C. ja Gavras, A., toimittajat, Springer, Lontoo, 2015, sivut 1-23.
- VFS15 Volk, M., Falk-Andersson, N., Sedlar, U., Hot to Elicit, Analyse and Validate Requirements for a Digital Health Solution. Teoksessa Requirements Engineering for Digital Health, Fricker, S. A., Thümmeler, C. ja Gavras, A., toimittajat, Springer, Lontoo, 2015, sivut 155-188.
- Vuo14 Vuosikatsaus 2013: reseptit sähköistyvät vauhdilla. Internetissä julkaistu artikkeli, Apteekkariliitto, 2014.  
[http://www.apteekkariliitto.fi/media/pdf/vuosikatsaus\\_2013.pdf](http://www.apteekkariliitto.fi/media/pdf/vuosikatsaus_2013.pdf). [20.11.2015]

## Liite 1

### Vaativuusmäärittelyn onnistumisen osatekijät [EIM95, 208]

- M1 = Ovatko käyttäjät tietoisia niistä liiketoiminnan muutoksista, joita tarvitaan suositellun ratkaisun käyttöönottamiseksi?
- M2 = Onko mallien ja tietojärjestelmän tavoitteiden väliset yhteydet kuvattu selkeästi sekä prosessien että tietojen kannalta?
- M3 = Kuvastuuko liiketoimintaprosessi selkeästi arkkitehtuurissa?
- M4 = Onko vaihtoehtoisen ratkaisut selvitetty perusteellisesti?
- M5 = Onko vaatimusmäärittelytyön tuotos-panos -suhde yhteneväinen organisaation omien tai muiden vastaavanlaisten organisaatioiden projekteihin?
- M6 = Paljonko vaatimusmäärittelydokumentaatioon tehdään muutoksia?
- M7 = Miten käyttäjät reagoivat kustannusarvioon?
- M8 = Ovatko käyttäjät halukkaita puolustamaan suositeltua ratkaisua johdolle?
- M9 = Onko kustannus-hyöty -analyysi kattava?
- M10 = Mikä osuus koko projektin kehittämiskustannuksista varataan vaatimusmäärittelytehtäviin?
- M11 = Miten paljon suoritteita jäi huomioimatta suositellun ratkaisun muodostamisessa ja kustannus-hyöty -analyysissä?
- M12 = Miten paljon enemmän organisaatio hyötyy suositellun ratkaisun käyttöönotosta verrattuna vaihtoehtoihin ratkaisuihin?
- M13 = Ovatko mallien ja avainasioiden väliset yhteydet selkeitä sekä prosessien että tietojen kannalta?
- M14 = Onko nykyinen tietojärjestelmä analysoitu kattavasti?
- M15 = Ollaanko varmoja, että valitut lähestymistavat mahdollistavat aineettomien hyötyjen saavuttamisen?
- M16 = Ovatko nykyisen järjestelmän heikkoudet ja vahvuudet selkeästi yhteydessä suositellun ratkaisun heikkouksiin ja vahvuuksiin?
- M17 = Missä määrin prosesseihin ja tietoihin liittyvät mallit noudattavat mallinnuksen sääntöjä?
- M18 = Missä laajuudessa avainkysymykset on pystytty ratkaisemaan?
- M19 = Ovatko käyttäjät ymmärtäneet, mitä tuleva järjestelmä tekee ja mitä se ei tee?
- M20 = Vallitseeko käyttäjien kesken yhteisymmärrys suositellusta ratkaisusta?
- M21 = Onko ylin johto vakuuttunut siitä, että tavoitellut hyödyt tulevat materialisoitumaan?
- M22 = Onko käyttäjien ja vaatimusmäärittelijöiden väliset suhteet hyvät?
- M23 = Ovatko käyttäjät hyväksyneet kaikki dokumentit?
- M24 = Onko käytettävissä oleva rahoitusmalli yhteneväinen suositeltuun ratkaisuun tarvittavan rahoitusmallin kanssa?
- M25 = Onko arkkitehtuuri yhteensopiva käyttäjien käyttötapojen kanssa?
- M26 = Onko organisaatiolla kyky tehdä tarvittavia muutoksia suositellun ratkaisun käyttöönottamiseksi?
- M27 = Sopiiko suositeltu ratkaisu yhteen organisaation strategian kanssa?
- M28 = Sopiiko suositeltu ratkaisu organisaation teknologialinjauksiin?
- M29 = Onko organisaatio halukas tekemään tarvittavat muutokset suositellun ratkaisun käyttöönottamiseksi?
- M30 = Onko kustannusarviot laadittu organisaation edellyttämällä tarkkuudella?

M31 = Tukeeko ylin johto tarvittavien muutosten tekemistä suositellun ratkaisun käyttöönottamiseksi?

M32 = Sopiiko suositeltu järjestelmäarkkitehtuuri yhteen organisaation arkkitehtuurin tai tietosuunnitelman kanssa?

M33 = Vastaavatko järjestelmän ensimmäisen julkaisun toiminnallisuudet käyttäjien odotuksiin?

M34 = Missä määrin kustannus-hyöty -analyysiesitys noudattelee organisaation kirjanpitoimenettelyitä?

## Liite 2

### Suomen sähköisen lääkemääräyksen toiminnallisuudet

[Säh15a, 16-22 mukaillen]

<i>Toiminnallisuus</i>	<i>Potilas-tietojärjestelmä</i>	<i>Apteekki-tietojärjestelmä</i>
<i>Asiakkaan lääkemääräystietojen ja -historian katselu</i>	X	X
<i>Lääkkeen määrääminen</i>	X	-
<i>Lääkemääräyksen korjaus</i>	X	X
<i>Lääkemääräyksen mitätöinti</i>	X	X
<i>Lääkemääräyksen toimitus</i>	-	X
<i>Lääketoimituksen korjaus</i>	-	X
<i>Lääketoimituksen mitätöinti</i>	-	X
<i>Lääkemääräyksen lukitus</i>	-	X
<i>Lääkemääräyksen lukituksen purku</i>	X	X
<i>Lääkemääräyksen varaus ja erityislupavaraus sekä niiden purku</i>	-	X
<i>Lääkemääräyksen toimitusvaraus ja sen purku</i>	-	X
<i>Annosjakelu ja sen purku</i>	-	X
<i>Lääkemääräyksen uusimispyynnön kirjaaminen</i>	X	X
<i>Lääkemääräyksen uusiminen</i>	X	-
<i>Uudistamiskiellon asettaminen</i>	X	-
<i>Asiakkaan sähköisten lääkemääräysten ja niiden toimitusten yhteenvedon tulostaminen</i>	X	X
<i>Potilasohjeen tulostaminen</i>	X	-
<i>Lääkemääräyksen toimitustiedon tulostaminen</i>	-	X
<i>Ulkomaan ostoa varten annettavan jäljennöksen tulostaminen</i>	X	-
<i>Ulkomaan matkaa varten annettavan jäljennöksen tulostaminen</i>	-	X

## Liite 3

### Karkeistettu vaatimusmäärittelymalli

#### 0. Vaatimusmäärittelyn esivalmistelut

- Laaditaan vaatimusmäärittelytyön strategia valitsemalla kyseiselle projektille keskeiset vaatimusmäärittelydokumentaation laatutavoitteet ja vaatimusmäärittelyn onnistumistekijät
- Varmistetaan riittävä panostus vaatimusmäärittelytyöhön (projektin kokonaispanostuksesta n. 15-30 % varattava vaatimusmäärittelyyn)
- Varmistetaan, että vaatimusmäärittelytyöhön on käytettävissä riittävästi osaavia henkilöresursseja (tilaajan edustajan lisäksi kehitystiimistä ainakin yksi kokenut, täysipäiväinen vaatimusmäärittelijä, lisäksi teknisiä suunnittelijoita, toteuttajia ja testaajia)

#### 1. VM1- iteraatiovariaatio (tilaajan näkökulma)

##### 1.1 Ongelmakentän ymmärtäminen ja vaatimusten kartutus

- Aloituskeskustelu (-t) tilaajan kanssa
  - Tavoitteena ymmärtää ongelmakenttä tilaajan näkökulmasta
  - Nykyratkaisun esittely
  - Nykyisten toimintaprosessien mallintaminen
  - Lait, asetukset, määräykset ja standardit, joita noudatettava
  - Muun taustamateriaalin tunnistaminen
- Taustamateriaalin kerääminen ja siihen tutustuminen
- WHY-WHAT-WHO -kysymyspatteriston esittäminen tilaajalle
  - Tavoitteena saada tilaajalta yksityiskohtaisempaa tietoa ongelmakentästä, tavoitteista ja sidosryhmistä
  - WHY
    - Ydinongelman selvittäminen tilaajan kannalta
  - WHAT
    - Tulevaan järjestelmään toivottavia toiminnallisuuksia ja ominaisuuksia
  - WHO
    - Tulevaan järjestelmään liittyvät sidosryhmät, tietojärjestelmät, laitteet, ohjelmistot
    - Sidosryhmien priorisointi niiden merkityksellisyyden mukaan
- Vastaavanlaisiin olemassa oleviin järjestelmiin ja ratkaisuihin tutustuminen
  - Tietoa eri ratkaisuvaihtoehdoista ja teknologioista + standardeista
- Alustavan vaatimusdokumentin kokoaminen vaiheen aikana, mm.
  - Toimiala ja ongelmakenttä
  - Olemassa oleva järjestelmä ja toimintaprosessit
  - Toimintaorganisaatio



- Tunnistetut ongelmat ja mahdollisuudet
- Tunnistetut ratkaisuvaihtoehdot
- Noudatettavat lait, asetukset, määräykset, standardit
- Alustavan vaatimusdokumentin liitteet:
  - Sanasto
  - Alustava vaatimusluettelo, jonne kirjataan tunnistetut vaatimusehdokkaat

## 1.2. Vaatimusten arviointi

- Alustavan vaatimusdokumentin katselmointipalaveri(-t) tilaajan kanssa
  - Tarkoituksena varmistaa, että vaatimusdokumentin kirjauksissa ei ole virheitä => tarvittaessa korjaukset / täydennykset
- Alustavan vaatimusluettelon katselmointi - ja arviointipalaveri (-t) tilaajan kanssa
  - Varmistetaan, että tunnistetut vaatimusehdokkaat on kirjattu oikein ja yksiselitteisesti
  - Varmistetaan, että tilaajan edustajat ovat yksimielisiä vaatimuksista => tarvittaessa neuvottelut ja muokkaus
  - Priorisoidaan vaatimukset tärkeysjärjestykseen
- Vaihtoehtoisten ratkaisuiden arviointipalaveri (-t) tilaajan kanssa
  - Kirjataan eri vaihtoehtoihin liittyvät riskit ja mahdollisuudet ylös
  - Priorisoidaan vaihtoehdot (panos-tuotos-riskit -näkökulmat)

## 1.3 Vaatimusten spesifiointi ja dokumentointi

- Ei-toiminnallisten vaatimusten jako alaluokkiin
- Jos vaatimusten kuvaamiseen on päätetty käyttää jotakin spesifikaatiokieltä, tehdään konvertointi
  - Jos on valittu formaali spesifikaatiokieli, on huolehdittava siitä, että kaikki vaatimusmäärittelyyn osallistuvat ymmärtävät sitä riittävällä tasolla
- Mallinnetaan alustavia ER- ja käyttötapauskaavioita sekä nykyjärjestelmästä että tulevasta järjestelmästä, ne liitteeksi vaatimusdokumenttiin
- Kaavioiden (ja muiden olennaisten materiaalien) katselmointi, jolla varmistetaan, että asiat on dokumentoitu oikein

## 1.4 Vaatimusten vahvistaminen ja laadunvalvonta

- Vaatimusdokumentaation validointi- ja verifiointikatselmointipalaveri(-t) tilaajan kanssa
  - Validoidaan vaatimukset ja muu dokumentaatio eli varmistetaan, että dokumentoidut vaatimukset täyttävät tilaajan todelliset tarpeet => tarvittaessa muokkaus
  - Verifioidaan vaatimukset ja muu dokumentaatio eli varmistetaan, ettei niissä ole keskinäisiä ristiriitoja => tarvittaessa muokkaus
- Dokumentaation läpikäynti tarkistuslistojen avulla

- Tuloksena vaatimusmäärittelydokumentaatio v1.0

## 2. VM2- iteraatiovariaatio (muiden sidosryhmien näkökulma)

### 2.1 Ongelmakentän ymmärtäminen ja vaatimusten kartutus

- Vaatimusmäärittelyn aloituskokous (tunnistettujen sidosryhmien edustajat)
  - Kehitysprojektin esittely (mitä ollaan tekemässä ja miksi)
  - Vaatimusmäärittelytyön perusteiden ja vaiheiden esittely
  - Työtapojen läpikäynti ja niistä sopiminen
  - Tähän mennessä kootun dokumentaation esittely
- Työpajatyöskentely pienryhmissä sidosryhmien edustajien kanssa (tai vaihtoehtoisesti kirjalliset kyselyt)
  - Nykyisten toimintaprosessien ja skenaarioiden mallintaminen sekä ongelma-kohtien kartoitus (demot)
  - Mahdollisesti tulevaan ratkaisuun liittyvien toimintaprosessien ja skenaarioiden mallintaminen, jos niihin tulossa muutoksia
  - WHY - WHAT - WHO -kysymyspatteristo
    - Uusien sidosryhmien priorisointi
- Tarvittaessa vaatimusdokumentin täydennys sidosryhmien näkemysten pohjalta
  - Muutokset versioitava ja mahdolliset ristiriidat korostettava arviointivaihetta varten
- Pienryhmissä (tai kyselyillä) toiminnallisten sekä ei-toiminnallisten vaatimusten kartutus ja uusien vaatimusehdokkaiden kirjaaminen vaatimusluetteloon
  - Muistettava dokumentoida myös vaatimuksen esittäjätaho

### 2.2 Vaatimusten arviointi

- Vaatimusdokumentin ja sen liitteiden sekä vaatimusluettelon katselmointi pienryhmissä
  - Varmistetaan, että vaatimusdokumenttiin ko. pienryhmän toimesta tehdyt päivitykset ovat oikeellisia (esimerkiksi prosessikuvaukset)
  - Varmistetaan, että pienryhmässä tunnistetut uudet vaatimusehdokkaat on muotoiltu oikein ja yksiselitteisesti
  - Varmistetaan, että sidosryhmän edustajat ovat yksimielisiä vaatimuksista => tarvittaessa neuvottelut ja muokkaus
- Vaatimusdokumentin katselmointipalaveri(-t) yhdessä kaikkien sidosryhmäedustajien kanssa
  - Tarkoituksena varmistaa, että vaatimusdokumentin kirjauksissa ei ole virheitä tai ristiriitoja => tarvittaessa korjaukset / täydennykset
- Vaatimusluettelon katselmointi - ja arviointipalaveri (-t) yhdessä kaikkien sidosryhmäedustajien kanssa
  - Katselmoidaan kaikki eri sidosryhmien tunnistamat vaatimukset => tarvittaessa vaatimusten yhdistely ja ristiriitojen eliminointi
  - Priorisoidaan vaatimukset tärkeysjärjestykseen
- Vaihtoehtoisten ratkaisuiden arviointipalaveri (-t) sidosryhmäedustajien kanssa

- Mahdollisten uusien ratkaisuvaihtoehtojen dokumentointi
- Päivitetään eri vaihtoehtoihin liittyviä riskejä ja mahdollisuuksia
- Priorisoidaan vaihtoehdot (panos-tuotos-riskit -näkökulmat)

## 2.3 Vaatimusten spesifiointi ja dokumentointi

- Vaatimusten luokittelu ja riippuvuuksien ym. metatiedon dokumentointi
- Jos vaatimusten kuvaamiseen on päätetty käyttää jotakin spesifikaatiokieltä, tehdään konvertointi
  - Jos on valittu formaali spesifikaatiokieli, on huolehdittava siitä, että kaikki vaatimusmäärittelyyn osallistuvat ymmärtävät sitä riittävällä tasolla
- Kaavioiden päivitys ja mallinnus, liitteeksi vaatimusdokumenttiin
  - Täydennetään alustavia ER- ja käyttötapauskaavioita sekä nykyjärjestelmästä että tulevasta järjestelmästä
  - Mallinnetaan tarvittavia tietovuo-, aktiviteetti-, viestiyhteys- tai tilakaavioita
- Kaavioiden katselmointi, jolla varmistetaan, että asiat on dokumentoitu oikein. Katselmointiin mukaan suunnittelijoita ja toteuttajia

## 2.4 Vaatimusten vahvistaminen ja laadunvalvonta

- Vaatimusdokumentaation validointi- ja verifiointikatselmointipalaveri(-t) sidosryhmäedustajien kanssa
  - Dokumentaation läpikäynti laatutavoitteiden näkökulmasta (täydellisyys, yhdenmukaisuus, sopivuus, yksiselitteisyys, mitattavuus, asianmukaisuus, kelpoisuus, ymmärrettävyys, hyvä rakenne, muunneltavuus, jäljitettävyys) => tarvittaessa vaatimusten / dokumentaation muokkaus
- Dokumentaation läpikäynti tarkistuslistojen avulla
- Tuloksena vaatimusmäärittelydokumentaatio v2.0
- Alustavien arkkitehtuurikuvausten, teknisten suunnitelmien sekä prototyypin laatiminen, jos se on tässä vaiheessa mielekästä

## 3. VM3- iteraatiovariaatio (jatkotyöstäminen)

### 3.1 Ongelmakentän ymmärtäminen ja vaatimusten kartutus

- Mahdollisen prototyypin (esim. käyttöliittymänäkymä tai jokin järjestelmän toiminnallinen komponentti) katselmointi, jos sellainen on saatavilla
- Skenaarioiden päivittäminen, jos niitä on tarpeen muuttaa
- Uusien vaatimusten kartutus, jos niitä ilmenee uuden ymmärryksen valossa
- Vaatimusdokumentin ja liitteiden päivittäminen tarpeellisin osin

### 3.2 Vaatimusten arviointi

- Vaatimusdokumentin ja -luettelon katselmointipalaveri(-t) yhdessä kaikkien sidosryhmäedustajien kanssa

- Tarkoituksena varmistaa, että vaatimusdokumentin kirjauksissa ei ole virheitä tai ristiriitoja => tarvittaessa korjaukset / täydennykset
- Katselmoidaan uudet vaatimukset => tarvittaessa vaatimusten yhdistely ja ristiriitojen eliminointi
- Priorisoidaan vaatimukset uudelleen tärkeysjärjestykseen
- Vaihtoehtoisten ratkaisuiden arviointipalaveri (-t) kaikkien sidosryhmäedustajien kanssa
  - Mahdollisten uusien ratkaisuvaihtoehtojen dokumentointi
  - Päivitetään eri vaihtoehtoihin liittyviä riskejä ja mahdollisuuksia
  - Priorisoidaan vaihtoehdot (panos-tuotos-riskit -näkökulmat)

### 3.3 Vaatimusten spesifiointi ja dokumentointi

- Jos vaatimusten kuvaamiseen on päätetty käyttää jotakin spesifikaatiokieltä, tehdään konvertointi
  - Jos on valittu formaali spesifikaatiokieli, on huolehdittava siitä, että kaikki vaatimusmäärittelyyn osallistuvat ymmärtävät sitä riittävällä tasolla
- Kaavioiden päivitys ja mallinnus, liitteeksi vaatimusdokumenttiin
  - Täydennetään mallinnettuja kaavioita sekä nykyjärjestelmästä että tulevasta järjestelmästä
- Kaavioiden (ja muiden olennaisten materiaalien) katselmointi, jolla varmistetaan, että asiat on dokumentoitu oikein

### 3.4 Vaatimusten vahvistaminen ja laadunvalvonta

- Vaatimusdokumentaation validointi- ja verifiointikatselmointipalaveri(-t) kaikkien sidosryhmäedustajien kanssa
  - Dokumentaation läpikäynti laatutavoitteiden näkökulmasta (täydellisyys, yhdenmukaisuus, sopivuus, yksiselitteisyys, mitattavuus, asianmukaisuus, kelpoisuus, ymmärrettävyys, hyvä rakenne, muunneltavuus, jäljitettävyys) => tarvittaessa vaatimusten / dokumentaation muokkaus
- Dokumentaation läpikäynti tarkistuslistojen avulla
- Tuloksena vaatimusmäärittelydokumentaatio v3.0... vX.0

## Liite 4

### Vaatimusmäärittelyn iteraatiovariaatiot vesiputousmallissa

